AD-A012 874

TURBULENT WAKES IN A STRATIFIED FLUID. PART II: USER'S SUMMARY GUIDE TO "WAKE" COMPUTER PROGRAM

Milton Teske

Aeronautical Research Associates of Princeton, Incorporated

Prepared for:

Office of Naval Research
Defense Advanced Research Projects Agency

August 1974

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER A.R.A.P. Report No. 226	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Turbulent Wakes in a Stratified Fluid Part II: User's Summary Guide to "WAKE" Computer Program		5. TYPE OF REPORT & PERIOD COVERED Final Report 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Milton Teske		8. CONTRACT OR GRANT NUMBER(s) N00014-72-C-0413
9. PERFORMING ORGANIZATION NAME AND ADDRESS Aeronautical Research Associates of Princeton, Inc. 50 Washington Road, Princeton, N. J. 08540		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS DARPA Order No. 1910
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Department of the Navy, Arlington, Va.		12. REPORT DATE August 1974 13. NUMBER OF PAGES 117
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Submarine wakes
Turbulence modeling
Stratified flow
Swirling wakes
Numerical fluid dynamics

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This document is a part of the final report on Contract N00014-72-C-0413 covering the period May 15, 1972, to June 30, 1974 and contains the user information needed to operate the WAKE program on the A.R.A.P. Digital Scientific Corporation META-4 computer system. Part I of this report summarizes the turbulent model derivation and verification, and the sensitivity of the wake collapse to changes in initial conditions.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

PRICES SUBJECT TO CHANGE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

A.R.A.P. REPORT NO. 226

TURBULENT WAKES IN A STRATIFIED FLUID

PART II: USER'S SUMMARY GUIDE TO "WAKE" COMPUTER PROGRAM

by

Milton Teske

Aeronautical Research Associates of Princeton, Inc.
50 Washington Road, Princeton, New Jersey 08540

August 1974

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. 1910

A.R.A.P. REPORT NO. 226

TURBULENT WAKES IN A STRATIFIED FLUID

PART II: USER'S SUMMARY GUIDE TO "WAKE" COMPUTER PROGRAM

by

Milton Teske

Program Code No.	438
Dates of Contract	15 May 1972 - 30 June 1974
Amount of Contract	$170,000.00
Principal Investigators	Coleman duP. Donaldson W. S. Lewellen
Scientific Officer	Director, Fluid Dynamics Programs Mathematics and Information Sciences Division Office of Naval Research

This work was supported by the
Defense Advanced Research Projects Agency of the Department
of Defense and was monitored by the Office of Naval Research
under Contract N00014-72-C-0413

Aeronautical Research Associates of Princeton, Inc.
50 Washington Road, Princeton, New Jersey 08540
609-452-2950

IW

August 1974

TABLE OF CONTENTS

1. INTRODUCTION

For the last two years A.R.A.P. has been developing a computer program capable of solving the three-dimensional steady flow problem of fluid motion in a stratified fluid. This work was accomplished in various stages, and is now assembled in one FORTRAN program called WAKE. This program resides on the A.R.A.P. computer system. This part of the final report summarizes the usage of WAKE and gives detailed explanations of its input requirements and output results. In Section 2 we briefly summarize the theoretical problem. In Section 3 we examine the numerical scheme used to solve the equations of motion. In Section 4 we detail the input/output specifications, including the structure of the important initial profile file. The Appendix gives a source listing of the entire WAKE program and its full subroutine complement. This guide is not intended as a full explanation of the WAKE program - only the FORTRAN listings can do that. Nor is it intended to demonstrate the conversion possibilities of WAKE to other computer facilities. Rather, with this guide a computer analyst unfamiliar with the WAKE program should be able to construct a needed set of initial profiles, grasp an overview of the numerical scheme and program structure, and produce suitably correct output from the A.R.A.P. computer facility in a fairly short period of time.

2. THE THEORETICAL PROBLEM

When a self-propelled body moves through a medium with a stratified density gradient, a wake is generated which expands behind the body as its potential energy increases. Far behind the body the potential and kinetic energies come into balance and the wake collapses. The heavier fluid, finding itself in a region of lighter background density, reestablishes the stable condition existing before passage of the body, but at the expense of the generation and transmittal of internal gravity waves. The theoretical prediction of this complex physical problem has been the subject of a great deal of study. The intent of the A.R.A.P. approach is to model the generated turbulence by the technique of invariant second-order closure and to follow the buildup and collapse phase through two Brunt-Vaisala (B.V.) periods of fluid motion. A more detailed explanation of the derivation of the equations is presented in Part 1 of this final report (ref. 1). For completeness we here present the derived, modeled, approximated and normalized equations as they stand prior to numerical solution.

For the turbulence q^2 $(= \overline{u'u'} + \overline{v'v'} + \overline{w'w'})$:

$$\frac{Dq^2}{Dt} = -\frac{2}{Fr^2}\,\overline{w'\rho'} - \frac{2q^2}{Re\lambda^2} + \frac{\partial}{\partial y}\left[\left(3v_c\,\frac{\overline{v'v'}}{q}\,\Lambda_y + \frac{1}{Re}\right)\frac{\partial q^2}{\partial y}\right]$$

$$+ \frac{\partial}{\partial z}\left[\left(3v_c\,\frac{\overline{w'w'}}{q}\,\Lambda_z + \frac{1}{Re}\right)\frac{\partial q^2}{\partial z}\right] - 2\,\overline{u_i'u_k'}\,\frac{\partial u_i}{\partial x_k} \qquad (2.1)$$

For the perturbation density $\hat{\rho}$:

$$\frac{D\hat{\rho}}{Dt} = + \frac{Pr}{Re} \nabla^2 \hat{\rho} - \frac{\partial \overline{v'\rho'}}{\partial y} - \frac{\partial \overline{w'\rho'}}{\partial z} + w \tag{2.2}$$

For the mean velocities u, v, and w:

$$\frac{Du}{Dt} = + \frac{1}{Re} \nabla^2 u - \frac{\partial \overline{u'v'}}{\partial y} - \frac{\partial \overline{u'w'}}{\partial z} \tag{2.3}$$

$$\frac{Dv}{Dt} = - \frac{\partial \pi}{\partial y} + \frac{1}{Re} \nabla^2 v - \frac{\partial \overline{v'v'}}{\partial y} - \frac{\partial \overline{v'w'}}{\partial z} \tag{2.4}$$

$$\frac{Dw}{Dt} = - \frac{\partial \pi}{\partial z} + \frac{1}{Re} \nabla^2 w - \frac{\partial \overline{v'w'}}{\partial y} - \frac{\partial \overline{w'w'}}{\partial z} - \frac{\hat{\rho}}{Fr^2} \tag{2.5}$$

For the scale length Λ:

$$\frac{D\Lambda}{Dt} = \nu_c \frac{\partial}{\partial x_i} \left(q\Lambda \frac{\partial \Lambda}{\partial x_i} \right) - s_1 \frac{\Lambda}{q^2} \overline{u_i' u_j'} \frac{\partial u_i}{\partial x_j} - s_2 \nu \frac{\Lambda}{\lambda^2}$$

$$- s_3 \delta_{3i} \frac{\Lambda}{q^2} \frac{\overline{u_i' \rho'}}{Fr^2} - s_4 \frac{1}{q} \left(\frac{\partial q\Lambda}{\partial x_i} \right)^2 \tag{2.6}$$

For perturbation pressure:

$$\nabla^2 \pi = - \frac{1}{Fr^2} \frac{\partial \hat{\rho}}{\partial z} - \frac{\partial^2 \overline{v'v'}}{\partial y^2} - 2 \frac{\partial^2 \overline{v'w'}}{\partial y \, \partial z} - \frac{\partial^2 \overline{w'w'}}{\partial z^2}$$

$$+ 2 \frac{\partial v}{\partial y} \frac{\partial w}{\partial z} - 2 \frac{\partial v}{\partial z} \frac{\partial w}{\partial y} - \frac{\partial}{\partial x} \left(\frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) \tag{2.7}$$

For the turbulent correlations:

$$0 = - \overline{u_i'u_k'} \frac{\partial u_j}{\partial x_k} - \overline{u_j'u_k'} \frac{\partial u_i}{\partial x_k} - \delta_{3i} \frac{\overline{u_j'\rho'}}{Fr^2} - \delta_{3j} \frac{\overline{u_i'\rho'}}{Fr^2}$$

$$- \frac{q}{\Lambda} \left(\overline{u_i'u_j'} - \frac{\delta_{ij} q^2}{3} \right) - 2(b-f) \frac{q^3}{3\Lambda} \delta_{ij}$$

$$0 = - \overline{u_i'u_j'} \frac{\partial \rho}{\partial x_j} - \overline{u_j'\rho'} \frac{\partial u_i}{\partial x_j} - \delta_{3i} \frac{\overline{\rho'^2}}{Fr^2} - \frac{Aq}{\Lambda} \overline{u_i'\rho'}$$

$$0 = - \overline{u_j'\rho'} \frac{\partial \rho}{\partial x_j} - sbq \frac{\overline{\rho'^2}}{\Lambda} \qquad (2.8)$$

A complete nomenclature may be found in Part 1. Restrictions and modifications, especially in regard to the Quasi-Equilibrium eqs. (2.8) for the turbulent correlations and the correction factor f, are also detailed in Part 1. The algebraic solution to eqs. (2.8), with the assumptions that the principle production gradients are in u and ρ, and density gradients in y are smaller than density gradients in z (for computation of f only) gives:

$$\overline{v'v'} = \frac{1 - 2b + 2f}{3} q^2$$

$$\overline{v'w'_s} = \frac{\overline{v'v'}}{c_1 Fr^2} \frac{\partial \hat{\rho}}{\partial y} \qquad \overline{v'w'} = \overline{v'w'_s} - \frac{q\Lambda}{3} \left(\frac{\partial w}{\partial y} + \frac{\partial v}{\partial z} \right)$$

$$\overline{v'\rho'} = - \frac{Fr^2 q}{\Lambda} \overline{v'w'_s}$$

$$\overline{u'v'} = - \frac{\Lambda}{q} \left(\overline{v'v'} \frac{\partial u}{\partial y} + \overline{v'w'_s} \frac{\partial u}{\partial z} \right)$$

$$\overline{w'\rho'} = \frac{1}{c_2}\left[\frac{q}{\Lambda}\left(\overline{v'v'}\,\frac{\partial\rho}{\partial z} + \overline{v'w'}_s\,\frac{\partial\hat{\rho}}{\partial y}\right) - \frac{\overline{v'\rho'}}{bsFr^2}\,\frac{\partial\hat{\rho}}{\partial y}\right]$$

$$\overline{w'w'} = \overline{v'v'} - \frac{2\Lambda}{qFr^2}\,\overline{w'\rho'}$$

$$\overline{u'u'} = q^2 - \overline{v'v'} - \overline{w'w'}$$

$$\overline{\rho'\rho'} = -\frac{\Lambda}{qbs}\left(\overline{v'\rho'}\,\frac{\partial\hat{\rho}}{\partial y} + \overline{w'\rho'}\,\frac{\partial\rho}{\partial z}\right)$$

$$\overline{u'\rho'} = \frac{1}{c_1}\left[\frac{\partial\rho}{\partial z}\left(\overline{v'w'}_s\,\frac{\partial u}{\partial y} + \overline{w'w'}\,\frac{\partial u}{\partial z}\right)\right.$$

$$\left. - \frac{q}{\Lambda}\left(\overline{u'v'}\,\frac{\partial\hat{\rho}}{\partial y} + \overline{v'\rho'}\,\frac{\partial u}{\partial y} + \overline{w'\rho'}\,\frac{\partial u}{\partial z}\right)\right]$$

$$\overline{u'w'} = -\frac{\Lambda}{q}\left[\frac{\overline{u'\rho'}}{Fr^2} + \overline{v'w'}_s\,\frac{\partial u}{\partial y} + \overline{w'w'}\,\frac{\partial u}{\partial z}\right] \tag{2.9}$$

With

$$c_1 = \frac{Aq^2}{\Lambda^2} - \frac{1}{Fr^2}\,\frac{\partial\rho}{\partial z}$$

$$c_2 = \frac{1}{Fr^2}\left(1 + \frac{1}{bs}\right)\frac{\partial\rho}{\partial z} - c_1$$

$$f = b - \left(\frac{1-2b}{3}\right)\left\{\frac{\Lambda^2}{q^2}\left(\frac{\partial u}{\partial y}\right)^2 - \frac{1}{c_1c_2}\left(\frac{\partial u}{\partial z}\right)^2\right.$$

$$\left.\left[\frac{A^2q^2}{\Lambda^2} + \left(1 - \frac{A}{bs}\right)\frac{1}{Fr^2}\,\frac{\partial\rho}{\partial z}\right] - \frac{1}{c_2Fr^2}\,\frac{\partial\rho}{\partial z}\right\}$$

and

$$\frac{\partial\rho}{\partial z} = \frac{\partial\hat{\rho}}{\partial z} + 1.$$

The solution procedure always involves the solution of eqs. (2.9) for the turbulent correlations and eq. (2.6) for the dynamic scale Λ. However, the solutions of the main variables q^2, $\hat{\rho}$, u, v, w, and π are governed by the regime of interest. Thus, only q^2, $\hat{\rho}$ and u are computed in a Phase I calculation where $Ri_o \lesssim 0.1$. Here $v = w = \pi = 0$ since the flow is far from collapse. In a Phase II calculation, we solve for q^2, $\hat{\rho}$, v, w, and π, assuming that $u = 0$; we are here restricted to flows for which $|u_{max}|/q_{max} \lesssim 0.1$. When the flow situation does not fit either condition, we calculate all the variables q^2, $\hat{\rho}$, u, v, w, and π in a Phase III calculation.

For the runs presented in Part I (ref. 1), we take our initial conditions on the main variables (when they are nonzero) to be those assembled in Table 2.1.

TABLE 2.1

TABULATION OF NONZERO INITIAL CONDITIONS

$$r = (y^2 + z^2)^{1/2}$$

q^2:
$$q^2 = \begin{cases} \dfrac{0.0108}{(1+r^2)^2} & \text{swirl and lift-force} \\ 0.0108 \exp[-0.69\, r^2] & \text{simple collapse} \end{cases}$$

$\hat{\rho}$:
$$\hat{\rho} = \begin{cases} z & z \leq 1 \\ z \exp[-2(r^2 - 1)] & z > 1 \end{cases}$$

u:
$$u = \begin{cases} 0.080\,(1-Cr^2) \exp[-Cr^2] \quad C = 3.125 & \text{momentumless} \\ -0.305 \exp[-5(r-0.2)^2] & \text{axial momentum} \end{cases}$$

v,w:
$$v = -\frac{Vz}{r} \qquad V = \begin{cases} \dfrac{(1 - \exp[-3r^2])(1.5 - r)^2}{15\, r} & r \leq 1.5 \\ 0 & r > 1.5 \end{cases}$$

$$w = \frac{Vy}{r}$$

for swirl cases

$$v = \gamma z \left(\frac{1}{E_+} - \frac{1}{E_-} \right) \qquad E_\pm = z^2 + (y \pm \ell)^2$$

$$w = \gamma \left(\frac{y-\ell}{E_-} - \frac{y+\ell}{E_+} \right) \qquad \ell = 0.4$$

$$\gamma = \pm 0.0033$$

for trim-force cases

π: $\pi = 0$ throughout

3. THE NUMERICAL APPROACH

Equations (2.1)-(2.8) are solved in the two-dimensional y-z plane by recasting them in finite difference form and applying the ADI technique of Peaceman, Rachford and Douglas (refs. 2 and 3). In this method we construct a two-dimensional grid in the y and z directions and march in the x direction as we follow the flow development downstream of the initial conditions. In the y and z directions the first and second derivatives are approximated by centered differences (spacing is variable), while forward differences are used in x (ref. 4). Inputted tolerance parameters control the size and speed of the marching direction, and the spacing variability and intensity.

At the beginning of a new step in Δx (perhaps the start of the run itself), the program performs an implicit sweep in one direction in $\Delta x/2$ and then sweeps in the other direction in $\Delta x/2$ to complete the ADI procedure. The initial sweeping direction alternates with each full step to unbias any solution near the edge of an expanding profile. These steps have used the <u>current</u> values of q^2, $\hat{\rho}$, u, v, w, and π, together with the gross scale Λ to step forward in x. Although the main variables are coupled by eqs. (2.1)-(2.5) and (2.8), we choose to use current values wherever necessary to decouple the equations completely. Solutions at the next x are then swept to compute maximum values, maximum changes, and various integrals of interest, along with the updated scales Λ_y and Λ_z. The next step Δx_{new} is computed based on the changes taken by the present step in relation to the maximum change permitted. The profiles are then swept again to update the pressure forcing function - the right side of eq. (2.7), and to compute the algebraic turbulence via eq. (2.8).

WAKE then calls the pressure iteration subroutine. It works as a miniature WAKE program by adding a $-\partial\pi/\partial t$ term to the left side of eq. (2.7) and performing iterations in π stepping from the current solution to the next steady state estimate. Appropriate output routines are then called, followed by the necessary set of routines that inspect the curvatures in the y and z directions, readjusting the profile again per inputted tolerance criteria. A new step is then taken.

The mainline program is called WAKE. The Appendix of this report contains a complete listing of WAKE and its subroutines; in-house disk I/O routines and other straightforward assembler routines are not included. These routines control the monitoring of the disk files storing the large array of data necessary to execute the calculation.

It may be worthwhile to realize that the file record length is 24 words, broken into four sections of six words each. The first section contains TV(6), where TV(1) = $\partial u/\partial y$; TV(2) = $\partial u/\partial z$; TV(3) = $\partial \rho/\partial y$; TV(4) = $\partial \rho/\partial z$; TV(5) = F; and TV(6) = π. The second section holds XV(6) at the present step value; the third holds the intermediate (first-sweep) values of XV(6); and the fourth section holds the new step values of XV(6). Here XV(1) = q^2, XV(2) = ρ; XV(3) = u; XV(4) = v; XV(5) = w; and XV(6) = Λ.

4. INPUT CONTROL

Initialization of a WAKE run requires the generation of an initial profile file (giving the desired initial conditions to the desired variables) and the input of an appropriately punched deck of computer cards. The initial file must be formed in a way totally compatible with the sample generation program shown at the end of the Appendix. It must be structured as follows.

First word:	the initial x position value.
Second word:	the number of points NY in the y direction and NZ in the z direction.
Next NY words:	the NY independent y values of the y direction mesh (monotonically increasing).
Next NZ words:	the NZ independent z values of the z direction mesh (monotonically increasing).
Next NY*NZ words:	the complete corresponding mesh values for the first initialized dependent variable. The file must contain these NY*NZ values in blocks of 10 y values at a time (for all z), until the last block contains enough values to reach NY. Thus, the blocks would be built as 10*NZ, 10*NZ, 5*NZ if NY=25.
Next NY*NZ words:	the second initialized variable.
.	
.	
.	
Next word:	- 1.0 to signal end-of-data.

With the disk buffering currently in operation, the file inversion program PBFFI must be called to invert the initial file four sectors at a time.

The input cards to the WAKE program (example copies are included in the Appendix) are as follows:

Card 1: INFLG, N, JOBE (3I4 format code). INFLG = 0 on a restart, = 1 on a start. N = 1 permits a new run to start before completing the current run; N = 0 does not. JOBE = value, the upper minute limit on current job execution time. If JOBE = 0, the program will not test for the job time.

Card 2: NRUNI, CMNT (I4, 19A4). If NRUNI = value, the current run is given this run number; if NRUNI = 0, the run number counter in the common file is updated by one. CMNT is a 19 element vector containing any desired comments (printed at the start of the run).

Card 3: NSTMX, NSTSI, XMIN, XMAX, DELX, MXHRS, LAMIN (2I4, 3F8.3, 2I4). NSTMX sets the maximum number of steps permitted for the run. NSTSI sets the initial step value, typically = 0. XMIN is the initial x value. XMAX is the maximum x value (the program terminates when reaching XMAX). DELX is the initial Δx spacing. MXHRS is the run time maximum hours (termination also). LAMIN = 0 signals a normal run with turbulence; LAMIN = 1 signals a run with laminar flow.

Card 4: MXRY, LYLFF, MXRZ, LZLFF (4I4). MXRY and MXRZ are the maximum number of mesh points possible in the y and z directions; the most possible currently is 40. LYLFF and LZLFF set the lower boundary flags, = 0 implies a free lower boundary; = - 1 implies a reflecting lower boundary.

Card 5: NRFV (14I4). NRFV (7,2) gives the reflection properties for the seven variables q^2, $\hat{\rho}$, u, v, w, Λ, π across the two axes. The first seven integers give the variable properties across the z axis (+ to - y); the second seven give their properties for + to - z across the y axis. The integer entry is 0 if the variable is zero at the axis, and 1 if the slope of the variable is zero across the axis.

Card 6: VSCAV (5F8.3). VSCAV is a five element array giving the scaling factors for q^2, $\hat{\rho}$, u, v, w; typically = 1.0.

Card 7: VWTFV (5F8.3) is a five element array giving the weighting factors for q^2, $\hat{\rho}$, u, v, w. Typically, weight = 1.0 is given to q^2, $\hat{\rho}$ and u; while = 0.0 is given to v and w (they do not control anything).

Card 8: ZEROV (7F8.3). ZEROV is a seven element array giving the edge values of q^2, $\hat{\rho}$, u, v, w, Λ and π, where q^2, u and π must = 0.0, but nonzero values are possible for $\hat{\rho}$, v and w. The edge value for Λ is formed internally within the program.

Card 9: EPSN, EPSX, EPSS, ECMN, ECMX (5F8.3). This card gives the run tolerances, and is used with the scaling and weighting factors to set running noise levels (EPSN times the appropriate VSCAV entry), maximum changes (EPSX), edge tolerances (EPSS), minimum curvature (ECMN), and maximum curvature (ECMX). Typically, EPSN = 0.001, EPSX = 0.05; EPSS = 0.001; ECMN = 0.02 and ECMX = 0.05.

Card 10: DXMAX, DXMIN, DXFMX, BUFAC, FCUR, DFFMN, DFFMX, DFRMX (8F8.3). This card sets the spacing in the three directions. In the marching direction x, DXMAX is the maximum step size permitted; DXMIN is the minimum size, = 0.00001; DXFMN is the maximum rate at which Δx can grow and the solution march downstream, = 1.5; BUFAC is the maximum factor (times EPSX) that a variable may change before the step size is reduced and the step tried again, = 2.0; FCUR multiplies ECMN and ECMX when curvatures require too many points, = 1.1; DFFMN is a minimum spacing factor (times the width of the profile) below which a point cannot be inserted, = 0.05; DFFMX is a maximum spacing factor above which a point must be added, = 0.05; and DFRMX is the optimum spacing ratio, traditionally set equal to 2.02.

Card 11: NIOLP, NIOPP, NFOLP, NTOPP, NTOLP, NSOLP, NIAAF, LOUT (8I4). The first six parameters control printout as a function of the number of steps taken by the solution. Thus, if NIOLP = 5, every fifth step will generate an intermediate line printer record of x, Δx, Λ_y, Λ_z, maximum values of all variables, and integrals of potential and kinetic energies, and momentum. NIOPP generates intermediate disk storage; NFOLP causes a print of normalized mesh output along a specifically inputted y and z value; NTOPP generates a disk save of the total variable profiles; NTOLP causes a line print/plot of the variable profiles; and NSOLP yields a print of nonnoise turbulence to the line printer. NIAAF sets the frequency of profile readjustment, = 2 from stability considerations. LOUT = -1 forces only a printer plot; = +1 forces only a numoer plot; = 0 permits both printer plot and full profile output to the line printer for every requested variable.

Card 12: XOUTV (10F8.3) gives specific x values at which total printouts and disk storage is requested (overrices the output control parameters on the previous card).

Card 13: NSTAT, NSTBC, NUFF, LIBUF, LFBUF, LTBUF, NSTPR, IDSV (14I4). NSTAT controls the solution configuration desired: NSTAT = 1 for the solution of q^2, $\hat{\rho}$, u; NSTAT = 2 for q^2, $\hat{\rho}$, v, w, π; and NSTAT = 3 for all variables. NSTBC prevents a backup and retry of a step for the initial number of steps (typically = 5) set by NSTBC. NUFF sets the smooth transition to an XOUTV value, = 4. LIBUF, LFBUF and LTBUF control intermediate, full, and total printouts on a backup (0 or 1). NSTPR is the number of steps in the pressure iteration loop, ≤ 5. IDSV is an array of seven elements giving the disk storage switch for each of the seven variables (= 0 if the variable is not to be stored, as must be the case currently for Λ; = 1 if the variable is to be stored for future plotting purposes).

Card 14: RE, G, PR, AS, A, B, BETA, C, S (9F8.3). This card gives the Reynolds number, (Froude number)$^{-2}$, Prandtl number, and turbulence model parameters currently set at AS = 2.5; A = 0.75; B = 0.125; BETA = 0.0; C = 0.1; S = 1.8.

Card 15: DC, U, RIS, DRDZ, SCALE, PCRIT, SCALM, SPORS, CPOR (9F8.3). DC = 0.3 is the diffusion coefficient; U is the free stream velocity = 1.0; RIS = 0.25 is the stability cutoff of the Richardson number; DRDZ is the constant background $\partial\rho_o/\partial z$. If DRDZ = 0.0, WAKE will read card 20. SCALE sets the initial value of the scale length. PCRIT = 0.005 is the value of the ratio of maximum change to maximum value below which the pressure iteration is said to converge (herewith, 0.5%). SCALM = 0.01 is the minimum scale length permitted. SPORS is the square of the distance to the wave-absorbing liner, normalized by Λ; and CPOR is the strength of the liner itself. The liner strength increases exponentially with a power of CPOR*(r^2 - SPORS).

Card 16: S1, S2, S5, S6, S7, S8, LSCAL (6F8.3, I4). The values of S1 - S8 set the constants in the dynamic scale equation. LSCAL = 0 means the scale equation is computed but does not influence the other variables. Currently, s_3 = S5; s_4 = S6 = S7 = $\frac{1}{2}$S8.

Card 17: QCUT, DIVP, DIVF, PNORM, PCUT, CMU, XFACT, XZERO, YOUT, ZOUT (10F8.3). QCUT = 0.001 is the factor times the current maximum value of q^2 that sets the computational noise level of q^2. If a local value of q^2 is below QCUT*q^2_{max}, that point is not considered when constructing the various integrals of interest. Re^{-1} is given the value of CMU, not RE^{-1}, at points beyond a squared distance, normalized by Λ, greater than PCUT (= 40). This procedure is a further attempt to stabilize the solution. DIVP = 0.1 is the fraction of DXMAX setting the minimum value of the step size Δx entering the calculation for the corrective divergence effect, the last term in eq. (2.7).

When the program Δx_{new} is below DIVP * DXMAX, the divergence correction may overwhelm the physics taking place, typically during the first few steps of a run. DIVF = 2.0 is the accepted value given to the effectiveness of the divergence term, so that $\partial/\partial x \approx - u/2\Delta x_{new}$. PNORM = 1.0 normalizes the pressure step-size factors that yield the step-size in the pressure iteration (ref. 4). The pressure step is $\Delta t(i)$ = PNORM * (YMAX-YMIN) * (ZMAX-ZMIN)/PN(i), where PN is (10.0, 30.0, 60.0, 100.0, 150.0). XFACT and XZERO give the rational distance downstream, so that (X/D) = XFACT * X + XZERO. Finally, YOUT and ZOUT are the values of y and z along which the full printout is computed for output display.

Card 18: FNAME (8A4) is the name of the disk files to be used during program execution, and are the working file (WAKWF), the gamma matrix file (WAKGM), the global intermediate save file (WAKGL) and the total profile plot save file (WAKPL). The common file (WAKCM)is entered into WAKE by a data statement.

Card 19: FILEN, ISTAT, ITYPE, LZMAX, LPRFL, IFULL (A4, A1, 5I4). This card controls the location of the initial profiles, and provides start conditions on various aspects of the WAKE program. When ITYPE = -1, FILEN is ignored and the initial profiles are sought in the entered profile plot file. When ITYPE = 0, FILEN gives the name of the file containing the initial profiles. For ITYPE > 0, these initial profiles are assumed to have been generated by the axisymmetric program JETMN. ISTAT is the counterpart of NSTAT and gives the run statistic for the initial profiles (1, 2 or 3). When LZMAX = 0 the program keeps track of the spread in Λ_z and prints and stores the running profiles at the first point of nonmonotonic behavior in Λ_z. Then LZMAX = 1 so that the procedure will not repeat. When LZMAX = - 1, the scales Λ_y and Λ_z are fixed at the SCALM value entered on card 15. LPRFL = -1 is the standard operating mode of the pressure loop, and signals to the body of the WAKE program

that the iteration is complete. When LPRFL = 0 on initialization, the pressure iterations are performed prior to the first step DELX in q^2, $\hat{\rho}$, u, v, w. IFULL = 1 forces a total printout to the line printer of the initial profiles; IFULL = 0 does not.

Card 20: ZL, DRDZH (2F8.3). When DRDZ = 0.0 on card 15, card 20 is read. Above z = 0.0 and below z = ZL, the background density gradient is set to DRDZH. For ZL < z < 0.0, DRDZ = -10^{-6}.

With the reading of these twenty cards, WAKE will begin computation. Beside the intermediate output described above, several integrals and functional values are printed every step by the pressure iteration and WAKSC subroutines. The heading above each value is chosen to give a quick identification of that value. The normalized divergence error indicates the closeness to which our calculation is maintaining $\nabla \cdot \overline{v} = 0$. The various kinetic and potential energy components give an indication of energy distribution within and outside the turbulent wake. The printout "error" of the pressure iteration should be disregarded since the pressure iteration scheme currently in use has evolved beyond the usefulness of "error". Otherwise, printout in WAKOT is explained when it is printed. The plot file is stored in the same manner that the initial profile file is constructed, only with multiple values of x, one set of data following the other.

The switches on the META-4 console permit on-going changes to the output scheme presented above. The switches used here are:

0: terminates a run during the pressure iteration loop
1: forces an intermediate line printer output
2: forces a full line printer output at YOUT and ZOUT
3: forces a total line printer output
4: forces an intermediate printout to the disk file

5: forces a total printout to the disk file

6: forces a total line printer output of the turbulence distributions

7: forces a printer picture of the main variables

13: terminates a run after a step in Δx (including the pressure iteration) has been completed

14: prints the full reflection vector on run initialization

There is, of course, no substitute for the actual program language. A copy of the FORTRAN logic is presented in the Appendix. The potential user must be warned that the weak link in the program is the capacity of the pressure iteration loop to maintain a consistent pressure solution behind the running solution of q^2, $\hat{\rho}$, u, v and w. The cross velocities v and w will react rapidly to a degenerating pressure field, but they react slowly to a build-up of the density field prior to collapse. The WAKE program runs slowly on the A.R.A.P. facility; a 24 hour run is standard when trying to extend collapse dynamics to one B.V. Since this program is a "one-man" operation, extreme care must be taken in the input of the data cards. Internal data consistency checks are simply not there.

5. REFERENCES

1. Lewellen, W. S., M. Teske and Coleman duP. Donaldson: "Turbulent Wakes in a Stratified Fluid," Part I, A.R.A.P. Report No. 226, 1974.

2. von Rosenberg, D. U.: Methods for the Numerical Solution of Partial Differential Equations, Elsevier Press (1969), p. 87.

3. Roache, P. J.: Computational Fluid Dynamics, Hermosa Publishers (1972), p. 180-185, 194-195.

4. Sullivan, R. D.: "A Program to Compute the Behavior of a Three-Dimensional Turbulent Vortex," ARL TR 74-0009 (1974), p. 15.

5. Carnahan, B., H. A. Luther and J. O. Wilkes: Applied Numerical Methods, Wiley and Sons (1969), pp. 452-453, 508.

APPENDIX

This Appendix contains the FORTRAN listing of the complete WAKE program as programmed at A.R.A.P. The first listing is the mainline WAKE followed by the listing of common, followed by the called subroutines in alphabetical order. WAKE consumes all 32 K of core storage, requiring every subroutine called by the mainline to be Localed, and subroutines within these routines (particularly WAKMY and WAKMZ) to be Secaled. Data and program storage would push beyond 200 K on a larger machine.

The last program in the Appendix gives a sample indication of the structure of a FORTRAN program generating the initial profile file. Following this program is a listing of an example input deck.

```
WAKE.S(0105)
*IOCS(2501 READER,1403 PRINTER)
**WAKE - STRATIFIED SUBMARINE WAKE MAINLINE
C
C  THIS IS THE STRATIFIED SUBMARINE WAKE MAINLINE
C
*COPY (CMWAK)
C
C  INITIATE THE RUN
C
      CALL WAKIN
      IF (LSTFL) 20,20,8
C
C  INTEGRATE THE EQUATIONS
C
5     GO TO (7,6),NSS
6     CALL WAKSZ
      GO TO (10,7),NSS
7     CALL WAKSY
      GO TO (6,10),NSS
C
C  INITIALIZE THE PROFILES
C
8     CALL WAKPI
C
C  COMPUTE AUXILIARY QUANTITIES
C
10    CALL WAKAC
C
C  CHECK SOLUTION VALUES
C
20    CALL WAKAL
      IF (LMLFL) 22,30,5
C
C  COMPUTE THE SUPEREQUILIBRIUM VALUES
C
22    CALL WAKSC
      IF (NSTAT-1) 24,30,24
C
C  COMPUTE PRESSURE SOLUTION
C
24    CALL WAKPC
C
C  OUTPUT RUN RESULTS
C
30    CALL WAKPP
      CALL WAKOT
      IF (LMLFL) 40,20,20
C
C  AUTOPOINT ADJUST PROFILES
C
40    IF (LIAAF) 50,20,50
50    IEND=NRST+JEND-1
      CALL WAKAA(D2DZM,ZOLDV,ZNEWV,NRNZV,NRST,IEND,MXRZ,LZFAF,NPNZ,
     1 LZLFF,2)
      CALL WAKAA(D2DYM,YOLDV,YNEWV,NRNYV,NYPS,NYPE,MXRY,LYFAF,NPNY,
     1 LYLFF,1)
      IF (LFCUR) 30,60,30
60    CALL WAKAZ
```

```
      GO TO 20
      END
CARD ID 0105  DB ADDR 30B0   DB CNT 0078
```

```
CMWAK.S(0105)
      DIMENSION ZMV(20),ZV(20),ZPV(20)
C
      COMMON IOLAY,IYPSN,NMOVE,NR,JOBE,NOUT,BUFR(1284),BUFS(1284)
      COMMON NWR,NWWZF,NPTSN,NCOMT,NWVEC,IBOT,ITOP
      COMMON NPOS,JZ,JY,LZFAF,LYFAF,LIAAF,NMAT,NVAR,NVART,NMR
      COMMON LIOLF,LIOPF,LFOLF,LTOPF,LTOLF,LSOLF,LSTFL,LMLFL
      COMMON LTRNF,LBURF,LPKRF,LOVCF,LOVFF,LFCUR,JERR
      COMMON YM,Y,YP,FMU,XK,CVV,BBETA,HBETA,BBS,Q,DXI,CWS,SCAL
      COMMON ZNEWV(40),YNEWV(40),NRNZV(40),NRNYV(40),NPNZ,NPNY
      COMMON IROWT,IROWA,IROWR,IROWG,MOOD,IMAPV(4),JSTAT
C
      COMMON ZM,IYPSM,IYPEM,XMMV(6),XMYV(6),XMPV(6)
      COMMON Z,IYPS,IYPE,XZMV(6),XV(6),XZPV(6)
      COMMON ZP,IYPSP,IYPEP,XPMV(6),XPYV(6),XPPV(6)
      COMMON TZMV(6),TV(6),TZPV(6),TPYV(6),TMYV(6)
      COMMON AV(6),GM(16),ROWB(18,40,3),ROWG(16,40)
      COMMON XMAT(6),YMAT(6),ZMAT(6),DVEC(24)
      COMMON ZA,IVECA,IVECB.IRFV(6,4)
      COMMON SLNID(3),GAMID(3),GLOID(3),PLTID(3),COMID(3)
C
      COMMON NSTMX,NSTST,XMIN,XMAX,NG,NP,NSS,NRUN,KLOK(6)
      COMMON NYPS,NYPE,MXRY,NRST,JEND,MXRZ,LSCAL,LENDF,DFFMN,CMU
      COMMON EPSN,EPSX,EPSS,ECMN,ECMX,VSCAV(5),VWTFV(5)
      COMMON DXMAX,DXMIN,DXFMX,BUFAC,DFFMX,DFRMX,DXSAV,X,XP,FCUR
      COMMON NPTS,NIOLP,NIOPP,NFOLP,NTOPP,NTOLP,NSOLP,NSTPR,LPRFL
      COMMON NIAAF,NSTAT,NSTBC,NOFF,LIBUF,LFBUF,LTBUF
      COMMON XOUTV(10),YOLDV(40),ZOLDV(40),IYPSV(40),IYPEV(40)
      COMMON RE,G,PR,DC,AS,S,A,B,BETA,C,U,RIS,DRDZ,PCRIT,SCALM
      COMMON LAMIN,MXHRS,LOUT,LZMAX,DX,CVS,D2DYM(40),D2DZM(40)
      COMMON EPSXV(5),EPSSV(5),ECMXV(5),XMOM,XPE,XKE,TURBX(10)
      COMMON FMAXV(7),YMAXV(7),ZMAXV(7),TMAXV(7),GMAXV(7)
      COMMON DEPST,DEPSI,SPORS,CPOR,PNORM,PCUT,QCUT,DIVP,DIVF
      COMMON S1,S2,S5,S6,S7,S8,XFACT,XZERO,SCALE,YSCAL,ZSCAL
      COMMON ZEROV(7),NRFV(7,2),IDSV(7),LDRDZ,DRDZL,DRDZH
      COMMON CMNT(19),FNAME(8),LYLFF,LZLFF,YOUT,ZOUT,SPACR(10)
C
      EQUIVALENCE (ZMV(1),ZM),(ZV(1),Z),(ZPV(1),ZP)
CART ID 0105  DB ADDR 4340   DB CNT 004A
```

```
WAKAA.S(0105)
**WAKAA - STRATIFIED SUBMARINE WAKE, AUTOPOINT ADJUST PROFILES
      SUBROUTINE WAKAA(D2DM,FOLDV,FNEWV,NRNV,ISTRT,IEND,MXP,LFAF,NPN,
     1 LFLFF,LL)
C
C  THIS SUBROUTINE IN THE WAKE PROGRAM DETERMINES THE NEW SET OF POINTS
C  NEEDED TO SATISFY ERROR TOLERANCES
C
      DIMENSION D2DM(40),FOLDV(40),FNEWV(40),NRNV(40),JPOS(2)
*COPY (CMWAK)
C
      DATA JPOS/1HY,1HZ/
C
1000  FORMAT(//33H CURVATURE TOLERANCES RELAXED IN ,A1,E15.5)
C
      IOLAY=11
      FNEWV(1)=FOLDV(ISTRT-1)
      NRNV(1)=ISTRT-1
      DFMIN=DFFMN*(FOLDV(IEND+1)-FOLDV(ISTRT-1))
      DFMAX=DFFMX*(FOLDV(IEND+1)-FOLDV(ISTRT-1))
      ISTRX=ISTRT+LFLFF
      RATIO=ECMX/ECMN
      FCURV=1.0
C
C  FORWARD PASS TO DETERMINE NEW PROFILE POINTS
C
100   JN=1
      DFNEW=1.0E10
      DO 128 J=ISTRX,IEND
      F=FOLDV(J)
      FP=FOLDV(J+1)
      IF (J-ISTRT) 1101,1102,1102
1101  FM=F+F-FP
      GO TO 1104
1102  FM=FOLDV(J-1)
1104  DFM=F-FM
      DFP=FP-F
      DFT=DFM+DFP
      I=NRNV(JN)
      IF (J-ISTRT) 123,1106,1106
1106  IF (I) 120,120,111
111   IF (I+1-J) 120,112,112
112   DFNEW=1.0E10
      IF (JN-1) 114,114,113
113   DFNEW=FNEWV(JN)-FNEWV(JN-1)
114   IF (DFT-DFMAX) 1145,1145,116
1145  IF (DFT*DFT*D2DM(J)*RATIO-FCURV) 115,115,116
115   IF (DFT/DFNEW-DFRMX) 126,126,116
116   IF (DFM/DFNEW-DFRMX) 117,117,119
117   IF (DFM-DFMAX) 118,118,119
118   IF (DFM*DFM*D2DM(J)-FCURV) 120,120,119
119   IF (DFM-DFMIN) 120,1195,1195
1195  JN=JN+1
      CALL WAKDS(D2DM(J-1),D2DM(J),DFM,DFNEW)
      FNEWV(JN)=FM+DFM
      NRNV(JN)=0
120   JN=JN+1
      FNEWV(JN)=F
      NRNV(JN)=J
```

```
      DFNEW=FNEWV(JN)-FNEWV(JN-1)
      IF (DFP/DFNEW-DFRMX) 123,123,125
123   IF (DFP-DFMAX) 124,124,125
124   IF (DFP*DFP*D2DM(J)-FCURV) 126,126,125
125   IF (DFP-DFMIN) 126,1255,1255
1255  JN=JN+1
      CALL WAKDS(D2DM(J),D2DM(J+1),DFP,DFNEW)
      FNEWV(JN)=F+DFP
      NRNV(JN)=0
126   IF (JN+4-MXP) 128,160,160
128   CONTINUE
      JN=JN+1
      FNEWV(JN)=FOLDV(IEND+1)
      NRNV(JN)=IEND+1
C
C  BACKWARD PASS OF NEW POINTS TO SATISFY RATIO CRITERION
C
      JS=0
130   JP=JN+JS
      FP=FNEWV(JN)
      FNEWV(JP)=FP
      NRNV(JP)=NRNV(JN)
      DO 141 J=3,JN
      I=JN+2-J
      F=FNEWV(I)
      NR=NRNV(I)
      JP=JP-1
      IF (JS) 132,132,131
131   FNEWV(JP)=F
      NRNV(JP)=NR
132   FM=FNEWV(I-1)
      DFP=FP-F
      DFM=F-FM
      IF (DFM/DFP-DFRMX) 140,140,134
134   NRM=NRNV(I-1)
      IF (NR*NRM) 140,140,135
135   DFO=DFM
      CALL WAKDS(D2DM(NR),D2DM(NRM),DFO,DFP)
      FO=F-DFO
      IF (NR-NRM-2) 137,136,137
136   NR=NR-1
      F=FOLDV(NR)
      IF (ABS(F-FO)-DFM/DFRMX) 138,138,137
137   NR=0
      F=FO
138   JP=JP-1
      IF (JS) 140,140,139
139   FNEWV(JP)=F
      NRNV(JP)=NR
140   FP=F
141   CONTINUE
C
C  CHECK FOR POINTS ADDED TO SATISFY RATIO CRITERION
C
      IF (JS) 142,142,150
142   JS=2-JP
      IF (JN+JS+4-MXP) 143,143,160
143   IF (JS) 150,150,130
C
```

```
C  CHECK WHETHER PROFILE MUST BE REORGANIZED
C
150    LFAF=0
       NR=ISTRT-1
       JN=JN+JS
       DO 156 J=1,JN
       NRN=NRNV(J)
       IF (NRN-NR) 151,155,151
151    LFAF=1
       GO TO 1565
155    NR=NR+1
156    CONTINUE
1565   NPN=JN
       IF (FCURV-1.0) 158,158,157
157    WRITE (NOUT,1000) JPOS(LL),FCURV
158    RETURN
C
C  MAXIMUM POINTS EXCEEDED ON AUTOPOINT ADJUSTMENT
C
160    FCURV=FCURV*FCUR
       IF (FCURV-1.0E10) 100,180,180
180    LFCUR=LL
       RETURN
       END
CART ID 0105  DB ADDR 2E50   DB CNT 011A
```

```
WAKAC.S(0105)
**WAKAC - STRATIFIED SUBMARINE WAKE, AUXILIARY CALCULATIONS
      SUBROUTINE WAKAC
C
C  THIS SUBROUTINE IN THE WAKE PROGRAM PERFORMS A NUMBER OF TASKS
C  AUXILIARY TO THE SOLUTION OF THE SYSTEM OF EQUATIONS
C
C  1) COMPUTES MAXIMUM ABSOLUTE VALUE FOR EACH VARIABLE AND Y AND Z
C     LOCATIONS OF SAME, MAXIMUM STEP CHANGE OF VARIABLES, AND
C     MAXIMUM X DERIVATIVES
C
C  2) COMPUTES FOR EACH Z ROW THE MAXIMUM SECOND DERIVATIVE
C     WITH RESPECT TO Z FOR ALL Y
C
C  3) COMPUTES FOR EACH Y COLUMN THE MAXIMUM SECOND DERIVATIVE
C     WITH RESPECT TO Y FOR ALL Z
C
C  4) CONSTRUCTS THE MESH VALUE PRINTOUT
C
C  5) COMPUTES THE APPROPRIATE LENGTH SCALES
C
      DIMENSION VALUE(40),TOTAL(7,40,2),PSIV(40)
*COPY (CMWAK)
      EQUIVALENCE (TOTAL(1,1,1),ROWG(1,1)),(VALUE(1),NRNZV(2))
      EQUIVALENCE (PSIV(1),ZNEWV(1)),(GM(1),TKE),(GM(2),VKE)
      EQUIVALENCE (GM(3),RPE),(GM(4),WPE),(GM(5),WKE)
      EQUIVALENCE (GM(6),AREA),(GM(7),PSIM),(GM(8),XLIFT)
C
C  ZERO PERTINENT VARIABLES
C
      IOLAY=5
      MOOD=1
      LFL=NPTSN-NSTST
      PLANE=1.0/FLOAT(LYLFF+2)/FLOAT(LZLFF+2)
      XMUM=0.0
      XPE=0.0
      TKE=0.0
      VKE=0.0
      RPE=0.0
      WPE=0.0
      WKE=0.0
      AREA=0.0
      PSIM=0.0
      XLIFT=0.0
      CALL SFVFL(0.0,PSIV,MXRY)
      CALL SFVFL(0.0,FMAXV,4*NVART)
      DO 40 I=1,2
      DO 35 JY=1,40
      CALL SFVMV(ZEROV,TOTAL(1,JY,I),NVART)
 35   CONTINUE
 40   CONTINUE
      CALL SFVFL(0.0,D2DYM,40)
      CALL SFVFL(0.0,D2DZM,40)
      CALL WAKSE(ECMX,ECMXV)
      NYPSX=NYPS+LYLFF
      DO 50 JY=NYPSX,NYPE
      IF (YOUT-YOLDV(JY)) 60,55,45
 45   IF (YOUT-YOLDV(JY+1)) 55,50,50
 50   CONTINUE
```

```
55    RATY=(YOUT-YOLDV(JY))/(YOLDV(JY+1)-YOLDV(JY))
C
C  INITIALIZE FOR PASSING THROUGH PROFILE
C
60    IROWA=2
      IROWR=4
      IF (LSTFL) 96,96,95
95    IROWR=2
96    NRSTX=NRST+LZLFF
      NREND=NRST+JEND-1
      NPOS=NRST-1
      CALL WAKRR(NRST-1,ZV)
      CALL WAKRR(NRST,ZPV)
      DO 250 NR=NRSTX,NREND
      PSIH=0.0
C
C  READ THREE SURROUNDING ROWS AND TEST FOR SELECTED Z IN DOMAIN
C
      IF (NR-NRST) 104,103,103
103   NPOS=NR
      CALL SFVMV(Z,ZM,NWWZF)
      CALL WAKMR(2,1)
      CALL SFVMV(ZP,Z,NWWZF)
      CALL WAKMR(3,2)
      CALL WAKRR(NR+1,ZPV)
      GO TO 105
104   ZM=Z+Z-ZP
105   DZM=Z-ZM
      DZP=ZP-Z
      DZT=ZP-ZM
      LYFAF=0
      IF (ZOUT-ZM) 107,1065,106
106   IF (ZOUT-Z) 1065,107,107
1065  LYFAF=1
      RATZ=(ZOUT-ZM)/(Z-ZM)
C
C  STEP THROUGH ALL Y POINTS COMPUTING AUXILIARY QUANTITIES
C
107   Y=YOLDV(IYPS-1)
      YP=YOLDV(IYPS)
      CALL WAKMP(IROWR,NR+1,IYPS-1,XPYV,1)
      CALL WAKMP(IROWR,NR,IYPS-1,XV,1)
      IF (NR-NRST) 108,109,109
108   CALL WAKRF(XPYV,XMYV,2)
      GO TO 110
109   CALL WAKMP(IROWR,NR-1,IYPS-1,XMYV,1)
110   IYPSX=IYPS+LYLFF
      IF (LFL) 112,112,111
111   CALL WAKMP(IROWA,NR,IYPS-1,AV,1)
112   CALL WAKMP(IROWR,NR+1,IYPS,XPPV,1)
      CALL WAKMP(IROWR,NR,IYPS,XZPV,1)
      IF (NR-NRST) 113,114,114
113   CALL WAKRF(XPPV,XMPV,2)
      GO TO 115
114   CALL WAKMP(IROWR,NR-1,IYPS,XMPV,1)
115   DO 200 JY=IYPSX,IYPE
      IF (JY-IYPS) 116,117,117
116   CALL WAKRF(XPPV,XPMV,1)
      CALL WAKRF(XZPV,XZMV,1)
```

```
      CALL WAKRF(XMPV,XMMV,1)
      YM=Y+Y-YP
      GO TO 130
117   IF (LFL) 120,120,119
119   CALL WAKMP(IROWA,NR,JY,AV,1)
120   CALL SFVMV(XPYV,XPMV,NMOVE)
      CALL SFVMV(XV,XZMV,NMOVE)
      CALL SFVMV(XMYV,XMMV,NMOVE)
      YM=Y
      Y=YP
      YP=YOLDV(JY+1)
      CALL WAKMP(IROWR,NR+1,JY+1,XPPV,1)
      CALL WAKMP(IROWR,NR,JY+1,XZPV,1)
      IF (NR-NRST) 125,128,128
125   CALL WAKRF(XPPV,XMPV,2)
      GO TO 130
128   CALL WAKMP(IROWR,NR-1,JY+1,XMPV,1)
130   DYM=Y-YM
      DYP=YP-Y
      DYT=YP-YM
C
C  PERFORM AUXILIARY COMPUTATIONS FOR EACH VARIABLE
C
      DO 145 I=1,NWVEC
      IF (LFL) 1406,1406,1402
1402  TEM=ABS(XV(I)-AV(I))
      IF (TEM-TMAXV(I)) 1406,1406,1403
1403  TMAXV(I)=TEM
1406  TEM=ABS(XV(I))
      IF (TEM-FMAXV(I)) 142,142,1407
1407  FMAXV(I)=TEM
      YMAXV(I)=Y
      ZMAXV(I)=Z
      IF (I-1) 1408,1408,142
1408  NRL=NR
      JYL=JY
142   IF (LIAAF) 143,145,143
143   IF (I-NVAR) 144,144,145
144   D2DY=((XZPV(I)-XV(I))/DYP-(XV(I)-XZMV(I))/DYM)/DYT/ECMXV(I)
      D2DZ=((XPYV(I)-XV(I))/DZP-(XV(I)-XMYV(I))/DZM)/DZT/ECMXV(I)
      TEM=ABS(D2DY)
      IF (TEM-D2DYM(JY)) 1447,1447,1446
1446  D2DYM(JY)=TEM
1447  TEM=ABS(D2DZ)
      IF (TEM-D2DZM(NR)) 145,145,1448
1448  D2DZM(NR)=TEM
145   CONTINUE
C
C  STORE MESH OUTPUT VALUES
C
      IF (LYFAF) 1515,1515,150
150   DO 151 I=1,NWVEC
      TOTAL(I,JY,1)=XMYV(I)+RATZ*(XV(I)-XMYV(I))
151   CONTINUE
1515  IF (YOUT-YM) 155,153,152
152   IF (YOUT-Y) 153,155,155
153   DO 154 I=1,NWVEC
      TOTAL(I,NR,2)=XZMV(I)+RATY*(XV(I)-XZMV(I))
154   CONTINUE
```

```
155   IF (JY-JYL) 158,156,158
156   VALUE(NR)=XV(1)
C
C  COMPUTATION OF INTEGRALS
C
158   CALL WAKLL(TEM,1)
      IF (TEM) 200,159,200
159   SUMF=PLANE*DYT*DZT
      LZFAF=0
      IF (NR-NRST) 1591,1592,1592
1591  SUMF=0.5*SUMF
      LZFAF=1
1592  IF (JY-NYPS) 1593,1594,1594
1593  SUMF=0.5*SUMF
      LZFAF=1
1594  XMOM=XMOM+XV(3)*(U+XV(3))*SUMF
      TKE=TKE+XV(1)*(U+XV(3))*SUMF
      TEM=(XV(4)-ZEROV(4))**2+(XV(5)-ZEROV(5))**2
      TEM=(XV(3)*XV(3)+TEM)*(U+XV(3))*SUMF
      RPE=RPE+XV(2)*XV(2)*SUMF
      VKE=VKE+TEM
      XLIFT=XLIFT+(XV(5)-ZEROV(5))*SUMF
      IF (XV(1)-QCUT*FMAXV(1)) 162,162,161
161   XPE=XPE+XV(2)*Z*SUMF
      AREA=AREA+SUMF
      GO TO 165
162   WKE=WKE+TEM
      TEM=XV(2)*XV(2)*SUMF
      XPE=XPE+TEM/2.0
      WPE=WPE+TEM
165   IF (LZFAF) 200,166,200
166   PSIH=PSIH+0.5*DYM*(XV(5)+XZMV(5)-2.0*ZEROV(5))
      PSIV(JY)=PSIV(JY)+0.5*DZM*(XV(4)+XMYV(4)-2.0*ZEROV(4))
      TEM=0.5*(PSIV(JY)-PSIH)
      IF (ABS(TEM)-ABS(PSIM)) 200,200,167
167   PSIM=TEM
200   CONTINUE
250   CONTINUE
      TKE=TKE/2.0
      VKE=VKE/2.0
      WKE=WKE/2.0
      XKE=TKE+VKE
      XLIFT=-XLIFT
      RPE=G*RPE/2.0
      WPE=G*WPE/2.0
      XPE=G*XPE
C
C  COMPUTATION OF GROSS SCALE LAMBDAS
C
      NPOS=NRL
      CALL WAKRR(NRL,ZV)
      Y=YOLDV(JYL)
      XV(1)=FMAXV(1)
      TEM=XV(1)/4.0
      IYPSX=JYL+1
      LFL=0
      DO 260 JY=IYPSX,IYPE
      XMYV(1)=XV(1)
      YM=Y
```

```
      CALL WAKMP(IROWR,NRL,JY,XV,1)
      Y=YOLDV(JY)
      IF (TEM-XV(1)) 256,252,252
252   IF (LFL) 260,254,260
254   YP=YM+(Y-YM)*(XMYV(1)-TEM)/(XMYV(1)-XV(1))
      LFL=1
      GO TO 260
256   LFL=0
260   CONTINUE
      EPSSV(3)=2.0*C*(YP-YOUT)
      NRSTX=NRL+1
      LFL=0
      DO 280 NR=NRSTX,NREND
      IF (TEM-VALUE(NR)) 276,272,272
272   IF (LFL) 280,274,280
274   ZP=ZOLDV(NR-1)+(ZOLDV(NR)-ZOLDV(NR-1))*(VALUE(NR-1)-TEM)/
     1 (VALUE(NR-1)-VALUE(NR))
      LFL=1
      GO TO 280
276   LFL=0
280   CONTINUE
      EPSSV(1)=2.0*C*(ZP-ZOUT)
      RETURN
      END
CART ID 0105  DB ADDR 4590   DB CNT 0206
```

```
WAKAL.S(0105)
**WAKAL - STRATIFIED SUBMARINE WAKE. AUXILIARY LOOP FOR EXECUTION
      SUBROUTINE WAKAL
C
C  THIS SUBROUTINE IN THE WAKE PROGRAM EXAMINES SOLUTION VALUES
C
*COPY (CMWAK)
C
      DATA JMSGF,JMSGR,JMSGB,JMSGS/2HST,2HRS,2HBU,2HOK/
C
1000  FORMAT(//47H FULL OUTPUT AT FIRST LOCAL MAXIMUM OF Z LAMBDA)
1001  FORMAT(//35H DX LESS THAN MINIMUM ALLOWED VALUE)
C
C  CHECK RUN POSITION
C
      IOLAY=6
      IF (LMLFL) 100,100,122
100   XP=X
      NPTSN=NPTS
      IF (LSTFL) 102,101,102
101   NPTSN=NPTS+1
      XP=X+DX
102   CALL WAKFS(NIAAF,LIAAF)
      LIOLF=0
      CALL WAKFS(NIOLP,LIOLF)
      CALL WAKFS(NIOPP,LIOPF)
      CALL WAKFS(NFOLP,LFOLF)
      CALL WAKFS(NTOPP,LTOPF)
      CALL WAKFS(NTOLP,LTOLF)
      CALL WAKFS(NSOLP,LSOLF)
      JSTAT=JMSGF
      IF (LSTFL) 103,110,105
103   IF (LPRFL) 104,170,170
104   LIOLF=1
      LIOPF=0
      LFOLF=0
      LTOLF=0
      LTOPF=0
      LSOLF=0
      JSTAT=JMSGR
      GO TO 170
105   LIOLF=1
      LFOLF=1
      LTOLF=IYPSN
      LSOLF=0
      LIOPF=NIOPP
      LTOPF=NTOPP
      LIAAF=NIAAF
      GO TO 164
C
C  SET UP FOR NEXT STEP
C
110   CALL WAKSE(EPSS,EPSSV)
      CALL WAKSE(EPSX,EPSXV)
      JSTAT=JMSGS
      DXSAV=DX
      DO 114 IXBRK=1,10
      XBRK=XOUTV(IXBRK)
      IF (X-XBRK) 112,114,114
```

```
112   DO 1122 I=1,NUFF
      TEM=FLOAT(I)
      IF (X+TEM*DX-XBRK) 1122,1124,1124
1122  CONTINUE
      GO TO 114
1124  DX=(XBRK-X)/TEM
      IF (I-1) 1126,113,1126
1126  XP=X+DX
      GO TO 116
113   XP=XBRK
1135  LIOLF=1
      LFOLF=1
      LIOLF=1
      LSOLF=1
      LIOPF=NIOPP
      LTOPF=NTOPP
      IF (LTRNF) 119,116,119
114   CONTINUE
116   IF (XP-XMAX) 118,117,117
117   LTRNF=1
      LENDF=1
      GO TO 1135
118   IF (NPTSN-NSTMX) 119,117,117
119   DXI=2.0/DX
C
C  INITIALIZE FOR CURRENT INTEGRATION STEP
C
      LMLFL=1
      IF (LBURF) 120,120,121
120   NSS=3-NSS
121   RETURN
122   LMLFL=0
      IF (LPKRF) 190,124,190
C
C  ERROR CHECK
C
124   CALL DVCHK(ISW)
      GO TO (1241,1242),ISW
1241  LDVCF=1
      LIOLF=1
1242  CALL OVERF(ISW)
      GO TO (1243,1244,1244),ISW
1243  LOVFF=1
      LIOLF=1
1244  IF (LDVCF+LOVFF) 175,130,175
C
C  CHECK FOR BACKUP AND DETERMINE NEW DX
C
130   LBURF=0
      DO 136 I=1,NVAR
      IF (TMAXV(I)-EPSXV(I)*BUFAC) 136,136,135
135   IF (NPTSN-NSTBC) 136,136,1355
1355  LBURF=I
136   CONTINUE
      DXN=DXSAV*ABS(DXFMX)
      DO 162 I=1,NVAR
      VMAX=GMAXV(I)
      IF (LBURF) 144,142,144
142   TEM=FMAXV(I)
```

```
      IF (TEM-VMAX) 1435,1435,143
143   GMAXV(I)=TEM
1435  VMAX=TEM
144   IF (VWTFV(I)) 162,162,145
145   IF (VMAX-VSCAV(I)*EPSN) 1455,146,146
1455  VMAX=VSCAV(I)*EPSN
146   AERR=ABS(EPSX)
      IF (EPSX) 147,148,148
147   AERR=AERR*VSCAV(I)
      GO TO 149
148   AERR=AERR*VMAX
149   AERR=AERR/VWTFV(I)
      TEM=TMAXV(I)/DX
      IF (TEM) 151,162,151
151   DXX=AERR/TEM
      IF (DXX) 162,162,160
160   IF (DXX-DXN) 161,162,162
161   DXN=DXX
162   CONTINUE
      DX=DXN
      IF (DX-DXMIN) 1620,1621,1621
1620  LIOLF=1
      LTRNF=1
      WRITE (NOUT,1001)
      GO TO 190
1621  IF (DXFMX) 163,1623,1622
1622  IF (DX-DXMAX) 163,163,1623
1623  DX=DXMAX
163   IF (LBURF) 1630,164,1630
1630  JSTAT=JMSGB
      LTRNF=1
      IF (DXSAV-DX*DXFMX) 1631,1632,1632
1631  DX=DXSAV/DXFMX
1632  LIOLF=0
      LFOLF=0
      LTOLF=0
      LSOLF=0
      LIOPF=0
      LTOPF=0
      IF (LIBUF) 1633,1634,1633
1633  LIOLF=1
1634  IF (LFBUF) 1635,1636,1635
1635  LFOLF=1
1636  IF (LTBUF) 1637,1638,1637
1637  LTOLF=1
1638  IF (LIBUF+LFBUF+LIBUF) 190,101,190
C
C  SUCCESSFUL INTEGRATION STEP
C
164   IF (LZMAX) 1646,1642,165
1642  IF (EPSSV(1)-ZSCAL) 1644,165,165
1644  LZMAX=1
      LIOLF=1
      LFOLF=1
      LTOLF=1
      LIOPF=NIOPP
      LTOPF=NTOPP
      WRITE (NOUT,1000)
      GO TO 165
```

```
1646  EPSSV(1)=SCALM
      EPSSV(3)=SCALM
165   ZSCAL=EPSSV(1)
      YSCAL=EPSSV(3)
      CALL SFVMV(FMAXV,GMAXV,NWVEC)
170   LMLFL=-1
175   CALL WAKCS(1,LIOLF)
      CALL WAKCS(2,LFOLF
      CALL WAKCS(3,LTOLF)
      CALL WAKCS(4,LIOPF)
      CALL WAKCS(5,LTOPF)
      CALL WAKCS(6,LSOLF)
190   CALL WAKCS(0,LTRNF)
      CALL WAKCS(13,LTRNF)
      RETURN
      END
CART ID 0105  DB ADDR 5880   DB CNT 0180
```

```
WAKAY.S(0105)
**WAKAY - STRATIFIED SUBMARINE WAKE, AUTOPOINT ADJUST Y
      SUBROUTINE WAKAY(NRX)
C
C  THIS SUBROUTINE IN THE WAKE PROGRAM REORGANIZES A ROW
C
      DIMENSION IYPNV(40)
*COPY (CMWAK)
      EQUIVALENCE (YOLDV(1),IYPNV(2))
C
C  MAKE WORKING COPY OF NRNYV
C
      NPOS=NRX
      CALL WAKRR(NRX,ZV)
      IF (IYPS) 160,160,100
100   IYS=0
      DO 108 IY=1,NPNY
      I=NRNYV(IY)
      IF (I) 106,106,101
101   IF (IYS) 102,102,104
102   IF (I-IYPS) 104,103,103
103   IYS=IY
104   IF (IYPE-I) 106,105,105
105   IYE=IY
106   IYPNV(IY)=I
108   CONTINUE
C
C  CHECK FOR MID-POINT ADDED ON EITHER END
C
      IF (IYS-1) 114,114,111
111   IF (NRNYV(IYS)-IYPS) 114,112,114
112   IF (NRNYV(IYS-1)) 113,113,114
113   IYS=IYS-1
114   IF (IYE-NPNY) 115,120,120
115   IF (NRNYV(IYE)-IYPE) 120,116,120
116   IF (NRNYV(IYE+1)) 117,117,120
117   IYE=IYE+1
C
C  CHECK WHETHER NRX ROW IS VOIDED BY ADJUSTMENT
C
120   IF (IYS-IYE) 121,121,1201
1201  IF (IYE-1) 121,121,1203
1203  IYS=IYE
C
C  FORWARD REORGANIZATION PASS
C
121   JYM=0
      IYS=MMAX(1,IYS-1)
      IYE=MMIN(NPNY,IYE+1)
      DO 150 IY=IYS,IYE
      JYT=NYPS+IY-2
      JYF=IYPNV(IY)
      I=IY
122   I=I+1
      IF (I-NPNY) 123,123,124
123   JYP=IYPNV(I)
      IF (JYP) 122,122,125
124   JYP=10000
125   IF (JYF) 126,126,132
```

```
126   IF (JYT-JYM) 150,150,127
127   IF (JYT-JYP) 128,150,150
128   CALL WAKIY(JYM,JYT,JYP,IY)
      GO TO 148
132   IF (JYT-JYF) 133,148,142
133   IF (JYT-JYM) 134,134,145
134   JYM=JYF
      GO TO 150
142   IF (JYT-JYP) 145,150,150
145   CALL SFVMV(ROWB(1,JYF,2),ROWB(1,JYT,2),NMOVE)
148   IYPNV(IY)=-JYT
      JYM=JYT
150   CONTINUE
C
C  BACKWARD REORGANIZATION PASS
C
      DO 154 JY=IYS,IYE
      JYT=NYPS+IYS+IYE-JY-2
      IY=IYS+IYE-JY
      JYF=IYPNV(IY)
      IF (JYF) 154,151,152
151   JYM=IABS(IYPNV(IY-1))
      JYP=IABS(IYPNV(IY+1))
      CALL WAKIY(JYM,JYT,JYP,IY)
      GO TO 153
152   CALL SFVMV(ROWB(1,JYF,2),ROWB(1,JYT,2),NMOVE)
153   IYPNV(IY)=-JYT
154   CONTINUE
      IYPS=NYPS+IYS-1
      IYPE=NYPS+IYE-3
160   CALL WAKWR(NRX,ZV)
      RETURN
      END
CART ID 0105  DB ADDR 32D0   DB CNT 00B6
```

```
WAKAZ.S(0105)
**WAKAZ - STRATIFIED SUBMARINE WAKE, AUTOPOINT ADJUST Z
      SUBROUTINE WAKAZ
C
C  THIS SUBROUTINE IN THE WAKE PROGRAM REORGANIZES THE SOLUTION FILE
C
*COPY (CMWAK)
C
C  CHECK PROFILE SHIFT
C
      IOLAY=12
      MOOD=-1
      IF (LZLFF) 101,102,102
101   NRSTN=2
      GO TO 108
102   NRSTN=(MXRZ-NPNZ)/2+2
      IF (NRST+(JEND-1)/2-MXRZ/2) 107,108,107
107   LZFAF=1
108   IF (LYLFF) 111,112,112
111   NYPSN=2
      GO TO 118
112   NYPSN=(MXRY-NPNY)/2+2
      IF (NYPS+(NYPE-NYPS)/2-MXRY/2) 117,118,117
117   LYFAF=1
118   IF (LYFAF) 1181,119,1181
C
C  SOLUTION PROFILE MUST BE REORGANIZED IN Y DIRECTION
C
1181  NYPS=NYPSN
      NYPE=NYPS+NPNY-3
      DO 1185 JZ=1,NPNZ
      NRF=NRNZV(JZ)
      IF (NRF) 1185,1185,1182
1182  CALL WAKAY(NRF)
1185  CONTINUE
      DO 1188 I=1,NPNY
      JZ=NYPS+I-2
      YOLDV(JZ)=YNEWV(I)
1188  CONTINUE
119   IF (LZFAF) 120,160,120
C
C  SOLUTION PROFILE MUST BE REORGANIZED IN Z DIRECTION
C
120   NRST=NRSTN
      JEND=NPNZ-2
C
C  FORWARD REORGANIZATION PASS
C
      NRM=0
      DO 150 JZ=1,NPNZ
      NRT=NRST+JZ-2
      NRF=NRNZV(JZ)
      J=JZ
122   J=J+1
      IF (J-NPNZ) 123,123,124
123   NRP=NRNZV(J)
      IF (NRP) 122,122,125
124   NRP=10000
125   IF (NRF) 126,126,132
```

```
126   IF (NRT-NRM) 150,150,127
127   IF (NRT-NRP) 128,150,150
128   CALL WAKIZ(NRM,NRT,NRP,JZ)
      GO TO 148
132   IF (NRT-NRF) 133,148,142
133   IF (NRT-NRM) 134,134,145
134   NRM=NRF
      GO TO 150
142   IF (NRT-NRP) 145,150,150
145   NPOS=NRF
      CALL WAKRR(NRF,ZV)
      NPOS=NRT
      CALL WAKWR(NRT,ZV)
148   NRNZV(JZ)=-NRT
      NRM=NRT
150   CONTINUE
C
C  BACKWARD REORGANIZATION PASS
C
      DO 154 JZ=1,NPNZ
      NRT=NRST+NPNZ-JZ-1
      IZ=NPNZ+1-JZ
      NRF=NRNZV(IZ)
      IF (NRF) 154,151,152
151   NRM=IABS(NRNZV(IZ-1))
      NRP=IABS(NRNZV(IZ+1))
      CALL WAKIZ(NRM,NRT,NRP,IZ)
      GO TO 153
152   NPOS=NRF
      CALL WAKRR(NRF,ZV)
      NPOS=NRT
      CALL WAKWR(NRT,ZV)
153   NRNZV(IZ)=-NRT
154   CONTINUE
      DO 156 I=1,NPNZ
      JZ=NRST+I-2
      ZOLDV(JZ)=ZNEWV(I)
156   CONTINUE
160   RETURN
      END
CART ID 0105  DB ADDR 2F70   DB CNT 00C4
```

```
WAKCL.S(0105)
**WAKCL - STRATIFIED SUBMARINE WAKE, COMPUTE LENGTH SCALES
      SUBROUTINE WAKCL(XXV,FTEM)
C
C  THIS SUBROUTINE IN THE WAKE PROGRAM COMPUTES THE VERTICAL AND
C  SCALE LENGTHS BY COMPARISON WITH THE RICHARDSON LENGTH
C
      DIMENSION XXV(6)
*COPY (CMWAK)
C
      SCAL=ZSCAL
      IF (FTEM) 103,140,103
103   TEM=SQRT(RIS*ABS(XXV(1)/FTEM)/G)
      IF (SCAL-TEM) 140,140,110
110   SCAL=TEM
      IF (SCAL-SCALM) 120,130,130
120   SCAL=SCALM
130   SCAL=SQRT(ABS(SCAL*ZSCAL))
140   IF (LSCAL) 150,160,150
150   SCALE=XXV(6)
      RETURN
160   SCALE=2.0*YSCAL*YSCAL*SCAL/(YSCAL*YSCAL+SCAL*SCAL)
      RETURN
      END
CART ID 0105  DB ADDR 3A90   DB CNT 002E
```

```
WAKCS.S(0105)
**WAKCS - STRATIFIED SUBMARINE WAKE, CHECK DATA SWITCH
      SUBROUTINE WAKCS(LL,LFL)
C
C  THIS SUBROUTINE IN THE WAKE PROGRAM CHECKS THE LL DATA SWITCH
C  FOR POSSIBLE OUTPUT TO LINE PRINTER OR DISK FILE
C
*COPY (CMWAK)
C
      CALL DATSW(LL,ISW)
      GO TO (10,20),ISW
10    LFL=1
      LIOLF=1
20    RETURN
      END
CART ID 0105  DB ADDR 2UE0   DB CNT 001C
```

```
WAKDG.S(0105)
**WAKDG - STRATIFIED SUBMARINE WAKE, DETERMINE DENSITY GRADIENT
      SUBROUTINE WAKDG(ZPOS)
C
C  THIS SUBROUTINE IN THE WAKE PROGRAM DETERMINES THE BACKGROUND
C  DENSITY GRADIENT WHERE APPLICABLE
C
*COPY (CMWAK)
C
      IF (LDRDZ) 10,100,10
10    IF (ZPOS) 20,30,30
20    IF (ZPOS-DRDZL) 30,30,25
25    DRDZ=-0.000001
      RETURN
30    DRDZ=DRDZH
100   RETURN
      END
CART ID 0105  DB ADDR 45A0   DB CNT 0020
```

```
WAKDS.S(0105)
**WAKDS - STRATIFIED SUBMARINE WAKE, OBTAIN DELTA SPACING RATIO
      SUBROUTINE WAKDS(CURVM,CURVP,DF,DFO)
C
C  THIS SUBROUTINE IN THE WAKE PROGRAM COMPUTES A NEW DF AS SOME
C  FRACTION OF THE OLD SO AS TO BALANCE THE THREE-POINT PROFILE
C
*COPY (CMWAK)
C
      DFRM=0.99*DFRMX
      IF (CURVP) 190,191,190
190   TEMB=SQRT(CURVM/CURVP)
      IF (TEMB-DFRM) 192,192,191
191   TEMB=DFRM
      GO TO 194
192   TEMA=1.0/DFRM
      IF (TEMB-TEMA) 193,194,194
193   TEMB=TEMA
194   DF=DF/(1.0+TEMB)
      IF (DF/DFO-DFRM) 196,196,195
195   DF=DFO*DFRM
196   RETURN
      END
CART ID 0105   DB ADDR 2700   DB CNT 002C
```

```
WAKEC.S(0105)
**WAKEC - STRATIFIED SUBMARINE WAKE, EDGE CONDITION CHECK
      SUBROUTINE WAKEC(LFL)
C
C  THIS SUBROUTINE IN THE WAKE PROGRAM TURNS THE IMPLICIT UPSWEEP
C  AND CHECKS FOR SATISFACTION OF THE BOUNDARY EDGE CONDITION
C
*COPY (CMWAK)
C
      LFL=1
      CALL SFVMV(AV,XMAT,NWVEC)
      DO 100 I=1,NWVEC
      XMAT(I)=XMAT(I)-GM(I)*ZEROV(I)
100   CONTINUE
      DO 110 I=1,NVAR
      IF (ABS(XMAT(I)-ZEROV(I))-EPSSV(I)) 110,110,120
110   CONTINUE
      CALL SFVMV(XMAT,AV,NWVEC)
      CALL WAKMP(IROWA,NR,JY,AV,2)
      RETURN
120   LFL=0
      RETURN
      END
CART ID 0105  DB ADDR 4310   DB CNT 002C
```

```
WAKFS.S(0105)
**WAKFS - STRATIFIED SUBMARINE WAKE, FLAG SET
      SUBROUTINE WAKFS(NPP,LFL)
C
C  THIS SUBROUTINE IN THE WAKE PROGRAM SETS THE OUTPUT FLAGS
C
*COPY (CMWAK)
C
      LFL=0
      IF (NPP) 102,102,100
100   IF (MOD(NPTSN,NPP)) 102,101,102
101   LFL=1
      LIOLF=1
102   RETURN
      END
CART ID 0105  DB ADDR 2E00   DB CNT 001C
```

```
WAKIN.S(0105)
**WAKIN - STRATIFIED SUBMARINE WAKE, INPUT CARDS AND START RUN
      SUBROUTINE WAKIN
C
C  THIS SUBROUTINE IN THE WAKE PROGRAM STARTS THE RUN
C
      DIMENSION ETID(3),LFLAG(6),COMN(2),FILEN(2),FILID(3)
      DIMENSION IDATE(4),JVCHV(5),FTID(3),HSCAL(2)
*COPY (CMWAK)
      EQUIVALENCE (LTRNF,LFLAG(1)),(AV(1),FILID(1))
C
      DATA COMN/4HWAKC,1HM/,JERRX/2H  /,BLANK/4H    /
      DATA HSCAL/3HOFF,3HON /
      DATA JVCHV/2HQQ,2HRO,2HU ,2HV ,2HW /
C
1000  FORMAT(18I4)
1001  FORMAT(7(A4,A1,3X))
1002  FORMAT(I4,19A4)
1003  FORMAT(2I4,3F8.3,2I4)
1004  FORMAT(6F8.3,I4)
1005  FORMAT(10F8.3)
1020  FORMAT(A4,A1,5I5)
2000  FORMAT(1H0,A4,A1,26H CANNOT BE FOUND OR OPENED)
2002  FORMAT(44H0JOB ABORT - WAKE PROGRAM IS IN RESTART MODE)
2006  FORMAT(45H1ARAP STRATIFIED SUBMARINE WAKE PROGRAM       ,19A4,
     1 7H   RUN ,I4)
2007  FORMAT(12H0RUN RESTART,11X,4A2)
2008  FORMAT(10H0RUN START,13X,4A2)
2009  FORMAT(20H0RUN SPECIAL RESTART,3X,4A2)
2010  FORMAT(/3X,9HMAX STEPS,3X,10HSTART STEP,4X,5HMIN X,7X,5HMAX X,
     1 4X,10HINITIAL DX,4X,7HMAX HRS,4X,8HLAM FLAG/
     2 I8,I12,4X,3E12.4,I8,I12)
2011  FORMAT(/2X,10HMAX Y SIZE,2X,10HLOW Y FLAG,2X,10HMAX Z SIZE,2X,
     1 10HLOW Z FLAG/I8,4I12)
2012  FORMAT(12X,7E12.4)
2013  FORMAT(14H0SCALE FACTORS,4X,5(A2,10X))
2014  FORMAT(15H0WEIGHT FACTORS,3X,5(A2,10X))
2015  FORMAT(/3X,9HNOISE MIN,2X,10HMAX CHANGE,2X,10HEDGE TOLER,3X,
     1 8HMIN CURV,4X,8HMAX CURV/5E12.4)
2016  FORMAT(/4X,6HMAX DX,6X,6HMIN DX,5X,9HDX FACTOR,3X,9HBU FACTOR,
     1 5X,4HCURV,6X,9HMIN SPACE,3X,9HMAX SPACE,5X,5HRATIO/8E12.4)
2017  FORMAT(15H0OUTPUT CONTROL,4X,9HINT PRINT,3X,8HINT DISK,3X,
     1 10HFULL PRINT,2X,10HTOTAL DISK,2X,11HTOTAL PRINT,2X,8HSE PRINT,
     2 3X,11HAUTO ADJUST,3X,7HPICTURE/12X,8I12)
2018  FORMAT(12H0X OUTPUT AT,10E12.4)
2022  FORMAT(/6X,2HRE,10X,1HG,11X,2HPR,8X,7HSMALL A,7X,1HA,11X,1HB,
     1 10X,4HBETA,9X,1HC,11X,1HS/9E12.4)
2023  FORMAT(/6X,2HDC,10X,1HU,11X,3HRI*,8X,4HDRDZ,8X,5HSCALE,4X,
     1 10HPRES ERROR,3X,9HMIN SCALE,4X,7HSPOR SQ,6X,4HCPOR/9E12.4)
2024  FORMAT(/6X,2HS1,10X,2HS2,10X,2HS5,10X,2HS6,10X,2HS7,10X,2HS8,
     1 6X,9HSCALE EQN/6E12.4,5X,A3)
2025  FORMAT(15H0SOLUTION FILES,4X,7HWORKING,4X,5HGAMMA,5X,6HGLOBAL,
     1 4X,4HPLOT,6X,6HCOMMON/20X,7(A4,A1,5X))
2031  FORMAT(18H0REFLECTION VECTOR,2X,14I6)
2032  FORMAT(12H0EDGE VALUES,6X,5(A2,10X),2HSL,10X,2HP )
2033  FORMAT(14H0CONTROL FLAGS,5X,9HTYPE STAT,4X,8HSTART-UP,5X,
     1 5HBREAK,6X,6HBU INT,6X,7HBU FULL,5X,8HBU TOTAL,4X,8HITERATES,
     2 4X,16HDISK SAVE VECTOR/12X,7I12,9X,7I2)
2034  FORMAT(17H0START CONDITIONS,3X,4HNAME,6X,5HVALUE,5X,6HOPTION,
```

```
     1 4X,6HLAMBDA,3X,8HPRESSURE,3X,6HOUTPUT/20X,A4,A1,I9,4I10)
2035  FORMAT(12HODRDZ CHANGE,2E12.4)
2040  FORMAT(/3X,8HQ CUTOFF,3X,10HDIVG PCENT,3X,9HDIVG FACT,4X,
     1 6HP NORM,5X,8HS CUTOFF,4X,9HCUTOFF MU,3X,8HX FACTOR,5X,
     2 6HX ZERO,5X,8HY OUTPUT,4X,8HZ OUTPUT/10E12.4)
C
C  RUN INITIALIZATION
C
      IOLAY=1
      CALL IIBFR(2,BUFR)
      CALL LETLI(ETID,BUFR)
      CALL IIBFR(2,BUFS)
      CALL LETLI(FTID,BUFS)
      NCOMT=419
      NWWZF=2
      NWR=24
      NMR=18
      NMOVE=12
      NMAT=16
      NINU=8
      NOUT=5
      NVAR=5
      NWVEC=6
      NVART=7
      IROWT=1
      IROWG=6
      IMAPV(1)=1
      IMAPV(2)=7
      IMAPV(3)=13
      IMAPV(4)=19
      CALL GDATE(IDATE)
C
C  OPEN COMMON SAVE DISK FILE AND READ IN CONTENTS
C
      L=1
      CALL LETLU(COMID,2,BUFR,ETID,COMN(1),-1,2,N)
      IF (N) 5,20,5
5     CALL SFVMV(COMN,FNAME,2)
10    WRITE (NOUT,2000) FNAME(L),FNAME(L+1)
      CALL EXIT
20    NRX=1
      CALL PBFDR(COMID,NRX,NCOMT,NSTST)
C
C  READ INITIAL INPUT CARD
C
      READ (NINU,1000) INFLG,N,JOBE
      LSTFL=1
      INFLG=INFLG-1
      IF (INFLG) 110,115,120
110   LSTFL=-1
      NSTST=NPTS
      GO TO 1211
115   IF (LENDF) 118,116,118
116   IF (N) 118,117,118
117   WRITE (NOUT,2002)
      CALL EXIT
118   LENDF=0
C
C  READ REMAINING INPUT CARDS
```

```
C
120    READ (NINU,1002) NRUNI,CMNT
       READ (NINU,1003) NSTMX,NSTSI,XMIN,XMAX,DELX,MXHRS,LAMIN
       READ (NINU,1000) MXRY,LYLFF,MXRZ,LZLFF
       READ (NINU,1000) NRFV
       READ (NINU,1005) VSCAV
       READ (NINU,1005) VWTFV
       READ (NINU,1005) ZEROV
       READ (NINU,1005) EPSN,EPSX,EPSS,ECMN,ECMX
       READ (NINU,1005) DXMAX,DXMIN,DXFMX,BUFAC,FCUR,DFFMN,DFFMX,DFRMX
       READ (NINU,1000) NIOLP,NIOPP,NFOLP,NTOPP,NTOLP,NSOLP,NIAAF,LOUT
       READ (NINU,1005) XOUTV
       READ (NINU,1000) NSTAT,NSTBC,NUFF,LIBUF,LFBUF,LTBUF,NSTPR,IDSV
       READ (NINU,1005) RE,G,PR,AS,A,B,BETA,C,S
       READ (NINU,1005) DC,U,RIS,DRDZ,SCALE,PCRIT,SCALM,SPORS,CPOR
       READ (NINU,1004) S1,S2,S5,S6,S7,S8,LSCAL
       READ (NINU,1005) QCUT,DIVP,DIVF,PNORM,PCUT,CMU,XFACT,XZERO,
     1 YOUT,ZOUT
       READ (NINU,1001) FNAME
C
C  OPEN REMAINING BASIC DISK FILES
C
1211  L=1
      CALL LETLU(SLNID,2,BUFS,FTID,FNAME(1),-1,2*NWR,N)
      IF (N) 10,1212,10
1212  L=3
      CALL LETLU(GAMID,2,BUFR,ETID,FNAME(3),-1,2*NMAT,N)
      IF (N) 10,1213,10
C
C  OPEN GLOBAL DISK FILE AND PROFILE DISK FILE
C
1213  IF (NIOPP) 1215,1215,1214
1214  L=5
      CALL LETLU(GLOID,2,BUFR,ETID,FNAME(5),-1,2,N)
      IF (N) 10,1216,10
1215  CALL SFVFL(BLANK,FNAME(5),2)
1216  IF (NTOPP) 1218,1218,1217
1217  L=7
      CALL LETLU(PLTID,2,BUFR,ETID,FNAME(7),-1,2,N)
      IF (N) 10,122,10
1218  CALL SFVFL(BLANK,FNAME(7),2)
122   IF (INFLG) 144,123,1224
C
C  SPECIAL RESTART WITH CURRENT WORKING FILE (NEW COMMON DATA)
C
1224  NSTST=NPTS
      LSTFL=-1
      GO TO 142
C
C  PROFILE INPUT INITIALIZATION
C
123   NSTST=NSTSI
      NPTS=NSTST
      IF (NRUNI) 125,125,124
124   NRUN=NRUNI-1
125   NRUN=NRUN+1
      IROWR=2
      ZSCAL=SCALE
      YSCAL=SCALE
```

```
      ZEROV(6)=2.0*SCALE
      X=XMIN
      IF (DELX) 130,130,129
129   DX=DELX
130   READ (NINU,1020) FILEN,LZFAF,LYFAF,LZMAX,LPRFL,IYPSN
      IF (LYFAF) 134,131,131
131   CALL LETLU(FILID,2,BUFR,ETID,FILEN(1),-1,2,N)
      IF (N) 132,140,132
132   WRITE (NOUT,2000) FILEN
      CALL EXIT
134   CALL SFVMV(FNAME(7),FILEN,2)
C
C  INITIALIZE FIXED PARAMETERS
C
140   NSS=1
      LIAAF=1
      DEPST=0.0
      DEPSI=0.0
      DXSAV=0.0
142   DO 143 I=1,6,2
      KLOK(I)=0
143   CONTINUE
      LDRDZ=0
      IF (DRDZ) 144,1435,144
1435  LDRDZ=1
      READ (NINU,1005) DRDZL,DRDZH
144   LSOLF=0
      DO 145 I=1,6
      LFLAG(I)=0
145   CONTINUE
      JERR=JERRX
      FMU=1.0/RE
      XK=PR/RE
      BBETA=1.0+2.0*B*BETA
      HBETA=1.5*BBETA
      BBS=1.0+BBETA/B/S
      CVV=(1.0-2.0*B*(1.0-BETA))/3.0/BBETA
C
C  REFLECTION PROPERTY CONDITION CONSTRUCTION
C
      DO 155 L=1,2
      DO 151 I=1,NWVEC
      IRFV(I,L)=NRFV(I,L)
151   CONTINUE
      DO 153 I=2,3
      DO 152 K=1,2
      N=2*(3-I)+K
      IRFV(N,L+2)=IABS(1-NRFV(I,L)-IABS(L-K))
152   CONTINUE
153   CONTINUE
      DO 154 N=5,NWVEC
      IRFV(N,L+2)=NRFV(7,L)
154   CONTINUE
155   CONTINUE
      CALL DATSW(14,ISW)
      GO TO (160,200),ISW
160   WRITE (NOUT,1000) IRFV
C
C  OUTPUT RUN PARAMETERS
```

```
C
200   WRITE (NOUT,2006) CMNT,NRUN
      IF (INFLG) 201,202,203
201   WRITE (NOUT,2007) IDATE
      GO TO 204
202   WRITE (NOUT,2008) IDATE
      GO TO 204
203   WRITE (NOUT,2009) IDATE
204   WRITE (NOUT,2010) NSTMX,NSTST,XMIN,XMAX,DX,MXHRS,LAMIN
      WRITE (NOUT,2011) MXRY,LYLFF,MXRZ,LZLFF
      WRITE (NOUT,2031) NRFV
      WRITE (NOUT,2013) JVCHV
      WRITE (NOUT,2012) VSCAV
      WRITE (NOUT,2014) JVCHV
      WRITE (NOUT,2012) VWTFV
      WRITE (NOUT,2032) JVCHV
      WRITE (NOUT,2012) ZEROV
      WRITE (NOUT,2015) EPSN,EPSX,EPSS,ECMN,ECMX
      WRITE (NOUT,2016) DXMAX,DXMIN,DXFMX,BUFAC,FCUR,DFFMN,DFFMX,DFRMX
      WRITE (NOUT,2017) NIOLP,NIOPP,NFOLP,NTOPP,NTOLP,NSOLP,NIAAF,LOUT
      WRITE (NOUT,2018) XOUTV
      WRITE (NOUT,2033) NSTAT,NSTBC,NUFF,LIBUF,LFBUF,LTBUF,NSTPR,IDSV
      WRITE (NOUT,2022) RE,G,PR,AS,A,B,BETA,C,S
      WRITE (NOUT,2023) DC,U,RIS,DRDZ,SCALE,PCRIT,SCALM,SPORS,CPOR
      WRITE (NOUT,2024) S1,S2,S5,S6,S7,S8,HSCAL(LSCAL+1)
      WRITE (NOUT,2040) QCUT,DIVP,DIVF,PNORM,PCUT,CMU,XFACT,XZERO,
     1 YOUT,ZOUT
      WRITE (NOUT,2025) FNAME,COMN
      IF (INFLG) 210,205,210
205   WRITE (NOUT,2034) FILEN,LZFAF,LYFAF,LZMAX,LPRFL,IYPSN
210   IF (LDRDZ) 215,220,215
215   WRITE (NOUT,2035) DRDZL,DRDZH
C
C  START RUN EXECUTION
C
220   LMLFL=-1
      NPTSN=NPTS
      RETURN
      END
CART ID 0105  DB ADDR 5570   DB CNT 0224
```

```
WAKIY.S(0105)
**WAKIY - STRATIFIED SUBMARINE WAKE, INSERT A Y POINT
      SUBROUTINE WAKIY(JYM,JYT,JYP,IY)
C
C  THIS SUBROUTINE IN THE WAKE PROGRAM INSERTS
C  A Y POINT BETWEEN TWO EXISTING ROW POINTS
C
*COPY (CMWAK)
C
      RATY=(YNEWV(IY)-YNEWV(IY-1))/(YNEWV(IY+1)-YNEWV(IY-1))
      DO 102 I=1,2
      CALL WAKMP(I,NPOS,JYM,XZMV,1)
      CALL WAKMP(I,NPOS,JYP,XZPV,1)
      DO 101 J=1,NWVEC
      XV(J)=XZMV(J)+RATY*(XZPV(J)-XZMV(J))
101   CONTINUE
      CALL WAKMP(I,NPOS,JYT,XV,2)
102   CONTINUE
      RETURN
      END
CART ID 0105  DB ADDR 2CC0   DB CNT 0026
```

```
WAKIZ.S(0105)
**WAKIZ - STRATIFIED SUBMARINE WAKE, INSERT A Z ROW
      SUBROUTINE WAKIZ(NRM,NRT,NRP,IZ)
C
C  THIS SUBROUTINE IN THE WAKE PROGRAM INSERTS
C  A Z ROW BETWEEN TWO EXISTING ROWS
C
*COPY (CMWAK)
C
C  FILL MINUS AND PLUS ROWS IN BUFFER
C
      NPOS=NRM+1
      CALL WAKRR(NRM,ZMV)
      NPOS=NRP-1
      CALL WAKRR(NRP,ZPV)
      Z=ZNEWV(IZ)
      RATZ=(Z-ZM)/(ZP-ZM)
      IF (IYPSM*IYPSP) 10,10,20
10    IYPS=MMAX(IYPSM,IYPSP)
      GO TO 30
20    IYPS=MMIN(IYPSM,IYPSP)
30    IYPE=MMAX(IYPEM,IYPEP)
      IYPSX=IYPS+LYLFF
C
C  INTERPOLATE FOR NRT ROW
C
      NPOS=NRT
      DO 114 JY=IYPSX,IYPE
      DO 112 I=1,2
      CALL WAKMP(I,NRT-1,JY,XMYV,1)
      CALL WAKMP(I,NRT+1,JY,XPYV,1)
      DO 111 J=1,NWVEC
      XV(J)=XMYV(J)+RATZ*(XPYV(J)-XMYV(J))
111   CONTINUE
      CALL WAKMP(I,NRT,JY,XV,2)
112   CONTINUE
114   CONTINUE
      CALL WAKWR(NRT,ZV)
      RETURN
      END
CART ID 0105  DB ADDR 2CF0   DB CNT 004E
```

```
WAKLL.S(0105)
**WAKLL - STRATIFIED SUBMARINE WAKE, LINER LOCATION CALCULATION
      SUBROUTINE WAKLL(TEM,LFL)
C
C  THIS SUBROUTINE IN THE WAKE PROGRAM DETERMINES WHETHER THE LINER
C  HAS BEEN REACHED AND SETS THE COEFFICIENT APPROPRIATELY
C
*COPY (CMWAK)
C
      TEM=0.0
      RR=Y*Y/YSCAL/YSCAL+Z*Z/ZSCAL/ZSCAL
      IF (RR-SPORS) 100,100,10
10    TEMM=CPOR*(RR-SPORS)
      GO TO (20,30),LFL
20    TEM=0.5*(EXP(TEMM)+EXP(-TEMM))-1.0
      RETURN
30    TEM=Y*(XV(4)-ZEROV(4))/YSCAL/YSCAL+Z*(XV(5)-ZEROV(5))/ZSCAL/ZSCAL
      TEM=CPOR*TEM*(EXP(TEMM)-EXP(-TEMM))
100   RETURN
      END
CART ID 0105  DB ADDR 2E20   DB CNT 0026
```

```
WAKMG.S(0105)
**WAKMG - STRATIFIED SUBMARINE WAKE, MOVE GAMMA ROW
      SUBROUTINE WAKMG(NRX,ZPOS,LFL)
C
C  THIS SUBROUTINE IN THE WAKE PROGRAM MOVES THE GAMMA MATRIX ROW
C  BETWEEN THE ROW BUFFER AND DISK FILE
C
      DIMENSION ZPOS(2)
*COPY (CMWAK)
C
      DATA JERRX/2HMG/
C
      IF (NRX) 100,100,10
10    IF (NRX-MXRZ) 20,20,100
20    CALL SFVMV(ZPOS,ZA,NWWZF)
      IYPSX=IVECA+LYLFF
      LL=NMAT*(IVECB-IYPSX+1)
      NRXX=(NRX-1)*MXRY+IYPSX
      GO TO (30,40),LFL
30    CALL PBFDR(GAMID,NRXX,LL,ROWG(1,IYPSX))
      RETURN
40    CALL PBFDW(GAMID,NRXX,LL,ROWG(1,IYPSX))
      RETURN
100   JERR=JERRX
      RETURN
      END
CART ID 0105  DB ADDR 2060   DB CNT 0032
```

```
WAKMP.S(0105)
**WAKMP - STRATIFIED SUBMARINE WAKE, MOVE A POINT
      SUBROUTINE WAKMP(IMAP,NRZ,NRY,VEC,LFL)
C
C  THIS SUBROUTINE IN THE WAKE PROGRAM MOVES SELECTED POINT
C  INFORMATION BETWEEN THE ROW OR GAMMA BUFFER AND VEC
C
      DIMENSION VEC(2)
*COPY (CMWAK)
C
      DATA JERRX/2HMP/
C
      J=NRZ-NPOS+2
      IF (J) 200,200,10
10    IF (J-3) 20,20,200
20    IF (IMAP-5) 60,200,30
30    GO TO (40,50),LFL
40    CALL SFVMV(ROWG(1,NRY),VEC,NMAT)
      RETURN
50    CALL SFVMV(VEC,ROWG(1,NRY),NMAT)
      RETURN
60    NRXX=IMAPV(IMAP)
      IF (IMAP-3) 90,70,80
70    IF (MOOD) 90,80,200
80    NRXX=NRXX-NWVEC
90    GO TO (100,110),LFL
100   CALL SFVMV(ROWB(NRXX,NRY,J),VEC,NWVEC)
      RETURN
110   CALL SFVMV(VEC,ROWB(NRXX,NRY,J),NWVEC)
      RETURN
200   JERR=JERRX
      RETURN
      END
CART ID 0105  DB ADDR 2DA0   DB CNT 0040
```

```
WAKMR.S(0105)
**WAKMR - STRATIFIED SUBMARINE WAKE, MOVE A BUFFER ROW
      SUBROUTINE WAKMR(NRF,NRT)
C
C  THIS SUBROUTINE IN THE WAKE PROGRAM MOVES THE CONTENTS OF THE
C  ROW BUFFER FROM ROW NRF TO ROW NRT
C
*COPY (CMWAK)
C
      DO 100 I=1,MXRY
      CALL SFVMV(ROWB(1,I,NRF),ROWB(1,I,NRT),NMR)
100   CONTINUE
      RETURN
      END
CART ID 0105  DB ADDR 2040   DB CNT 001A
```

```
WAKMV.S(0105)
**WAKMV - STRATIFIED SUBMARINE WAKE, SUPEREQUILIBRIUM MATRIX VALUES
      SUBROUTINE WAKMV(XXV,TTV,T)
C
C  THIS SUBROUTINE IN THE WAKE PROGRAM COMPUTES THE SUPEREQUILIBRIUM
C  MATRIX VALUES NEEDED FOR THE IMPLICIT UPSWEEP CALCULATION
C
      DIMENSION XXV(6),TTV(6),T(10)
*COPY (CMWAK)
C
      IF (XXV(1)) 100,100,101
100   CALL SFVFL(0.0,T,10)
      CVS=0.0
      CWS=0.0
      RETURN
101   Q=SQRT(XXV(1))
      FTEM=TTV(4)+DRDZ
      CALL WAKCL(XXV,FTEM)
      IF (LAMIN) 100,102,100
102   BQL=BBETA*Q/SCALE
      C1=A*BQL*Q/SCALE-G*FTEM
      C2=BBS*G*FTEM-C1
      TEM=TTV(1)*TTV(1)*SCALE/Q/BQL-BBETA*G*FTEM/C2
     1 -TTV(2)*TTV(2)*BBETA*(G*FTEM*(1.0-A/B/S)+A*A*XXV(1)
     2 /SCALE/SCALE)/C1/C2
      CVS=CVV+(B-CVV*TEM)/(HBETA+TEM)
      CALL WAKTC(CVS,1.0,IVECA)
      CVW=CVS/C1
      C4=CVW*G*TTV(3)
      T(7)=-BQL*CVW*XXV(1)
      C3=CVS*DRDZ+C4*BBS*TTV(3)
      T(4)=CVS*XXV(1)*BQL/C2
      T(3)=C3*BQL/C2
      CWS=CVS-2.0*G*(C3+CVS*TTV(4))/C2
      CALL WAKTC(CWS,1.0-CVS,IVECB)
      DUR=(FTEM*CWS*XXV(1)+C4*TTV(3)*XXV(1)
     1 -BQL*(T(3)*XXV(1)+T(4)*TTV(4)))/C1
      T(2)=-(G*DUR+CWS*XXV(1))/BQL
      T(6)=-CVS*XXV(1)/BQL
      T(5)=-C4*TTV(2)/BQL
      CUR=(FTEM*C4*TTV(1)+TTV(3)*TTV(1)*CVS*(1.0
     1 +BQL*BQL/C1))/C1
      T(1)=-(G*CUR+C4*TTV(1))/BQL
      T(8)=CVS
      T(9)=CWS
      T(10)=C4
      RETURN
      END
 CART ID 0105  DB ADDR 41C0   DB CNT 005E
```

```
WAKMY.S(0105)
**WAKMY - STRATIFIED SUBMARINE WAKE, Y IMPLICIT MATRIX COEFFICIENTS
      SUBROUTINE WAKMY
C
C  THIS SUBROUTINE IN THE WAKE PROGRAM COMPUTES THE X,Y,Z AND D MATRIX
C  COEFFICIENTS FOR THE Y IMPLICIT UPSWEEP
C
      DIMENSION T(10,5)
*COPY (CMWAK)
      EQUIVALENCE (T(1,1),ZNEWV(1))
C
      FPRZ(ARGM,ARG,ARGP)=FZM*(ARGM+FZS*ARG-FZR*ARGP)
      FPRY(ARGM,ARG,ARGP)=FYM*(ARGM+FYS*ARG-FYR*ARGP)
      SPRZ(ARGM,ARG,ARGP)=SZM*ARGM+SZ*ARG+SZP*ARGP
      SPRY(ARGM,ARG,ARGP)=SYM*ARGM+SY*ARG+SYP*ARGP
C
C  COMPUTE SPACING FACTORS
C
      DZM=Z-ZM
      DZP=ZP-Z
      DZT=DZM+DZP
      FZM=-DZP/DZM/DZT
      FZP=DZM/DZP/DZT
      FZ=-FZM-FZP
      TEM=DZM/DZP
      FZR=TEM*TEM
      FZS=FZR-1.0
      SZM=2.0/DZM/DZT
      SZP=2.0/DZP/DZT
      SZ=-SZM-SZP
C
      DYM=Y-YM
      DYP=YP-Y
      DYT=DYM+DYP
      FYM=-DYP/DYM/DYT
      FYP=DYM/DYP/DYT
      FY=-FYM-FYP
      TEM=DYM/DYP
      FYR=TEM*TEM
      FYS=FYR-1.0
      SYM=2.0/DYM/DYT
      SYP=2.0/DYP/DYT
      SY=-SYM-SYP
C
      CALL SFVFL(0.0,XMAT,NWVEC)
      CALL SFVFL(1.0,YMAT,NWVEC)
      CALL SFVFL(0.0,ZMAT,NMOVE)
C
C  COMPUTE MULTIPLICATIVE FACTORS
C
      CALL WAKDG(ZM)
      CALL WAKMV(XMYV,TMYV,T(1,5))
      CALL WAKDG(ZP)
      CALL WAKMV(XPYV,TPYV,T(1,4))
      CALL WAKDG(Z)
      CALL WAKMV(XZMV,TZMV,T(1,3))
      CALL WAKMV(XZPV,TZPV,T(1,2))
      CALL WAKMV(XV,TV,T(1,1))
      TMU=FMU
```

```
      IF (SPORS) 40,40,20
20    TEM=Y*Y/YSCAL/YSCAL+Z*Z/ZSCAL/ZSCAL
      IF (TEM-PCUT) 40,40,30
30    TMU=CMU
40    CL=2.0*(AS*TMU/SCALE+B*Q)/SCALE
C
C  COMPUTE DERIVATIVE FACTORS
C
      DXX=DXI*(U+XV(3))
      CALL WAKLL(TEMK,1)
      ADM=FYM*XZMV(4)
      AD=FY*XV(4)
      ADP=FYP*XZPV(4)
      SL=SCALE/3.0
      FQY=FPRY(SQRT(XZMV(1)),Q,SQRT(XZPV(1)))
      FQZ=FPRZ(SQRT(XMYV(1)),Q,SQRT(XPYV(1)))
      FKZ=FPRZ(XMYV(1),XV(1),XPYV(1))
      FRZ=FPRZ(XMYV(2),XV(2),XPYV(2))
      FUY=FPRY(XZMV(3),XV(3),XZPV(3))
      FUZ=FPRZ(XMYV(3),XV(3),XPYV(3))
      FVY=FPRY(XZMV(4),XV(4),XZPV(4))
      FVZ=FPRZ(XMYV(4),XV(4),XPYV(4))
      FWY=FPRY(XZMV(5),XV(5),XZPV(5))
      FWZ=FPRZ(XMYV(5),XV(5),XPYV(5))
      FLY=FPRY(XZMV(6),XV(6),XZPV(6))
      FLZ=FPRZ(XMYV(6),XV(6),XPYV(6))
      SPLY=3.0*DC*YSCAL*CVS*Q+TMU
      FPLY=3.0*DC*YSCAL*CVS*FQY
      SPLZ=3.0*DC*SCAL*CWS*Q+TMU
      FPLZ=3.0*DC*SCAL*CWS*FQZ
      PROM=FUY*T(5,1)+FUZ*T(1,1)+FVY*T(8,1)+(FWY+FVZ)*T(10,1)+FWZ*T(9,1)
      PROP=FUY*FUY*T(6,1)+FUZ*FUZ*T(2,1)-Q*SCALE*(FWY+FVZ)**2/3.0
C
C  COMPUTE MATRIX ELEMENTS FOR QQ
C
      TEMD=SPLZ*SPRZ(XMYV(1),XV(1),XPYV(1))+FPLZ*FKZ
      TEMC=FPRZ(XMYV(1)*XMYV(5),XV(1)*XV(5),XPYV(1)*XPYV(5))
      XMAT(1)=-SPLY*SYM-FPLY*FYM+ADM
      YMAT(1)=DXX-SPLY*SY-FPLY*FY+AD+CL+2.0*(PROM+G*T(3,1))
      ZMAT(1)=-SPLY*SYP-FPLY*FYP+ADP
      DVEC(1)=DXX*XV(1)-2.0*(PROP+G*FRZ*T(4,1))+TEMD-TEMC
C
C  COMPUTE MATRIX ELEMENTS FOR RHO
C
      TEMM=T(7,1)-XK
      TEMP=FPRY(T(7,3),T(7,1),T(7,2))
      TEMD=(T(4,1)-XK)*SPRZ(XMYV(2),XV(2),XPYV(2))
     1 +FPRZ(T(4,5),T(4,1),T(4,4))*FRZ
      TEMQ=FPRZ(T(3,5)*XMYV(1),T(3,1)*XV(1),T(3,4)*XPYV(1))
      TEMC=FPRZ(XMYV(2)*XMYV(5),XV(2)*XV(5),XPYV(2)*XPYV(5))
      XMAT(2)=TEMM*SYM+TEMP*FYM+ADM
      YMAT(2)=DXX+TEMM*SY+TEMP*FY+AD
      ZMAT(2)=TEMM*SYP+TEMP*FYP+ADP
      DVEC(2)=DXX*XV(2)-TEMD-TEMQ-DRDZ*XV(5)-TEMC
      IF (NSTAT-2) 50,60,50
C
C  COMPUTE MATRIX ELEMENTS FOR U
C
50    TEMM=T(6,1)-TMU
```

```
      TEMP=FPRY(T(6,3),T(6,1),T(6,2))
      TEMD=(T(2,1)-TMU)*SPRZ(XMYV(3),XV(3),XPYV(3))
     1 +FPRZ(T(2,5),T(2,1),T(2,4))*FUZ
      TEMQ=FPRY(T(5,3)*XZMV(1),T(5,1)*XV(1),T(5,2)*XZPV(1))
     1 +FPRZ(T(1,5)*XMYV(1),T(1,1)*XV(1),T(1,4)*XPYV(1))
      TEMC=FPRZ(XMYV(3)*XMYV(5),XV(3)*XV(5),XPYV(3)*XPYV(5))
      XMAT(3)=TEMM*SYM+TEMP*FYM+ADM
      YMAT(3)=DXX+TEMM*SY+TEMP*FY+AD
      ZMAT(3)=TEMM*SYP+TEMP*FYP+ADP
      DVEC(3)=DXX*XV(3)-TEMD-TEMQ-TEMC
      IF (NSTAT-2) 70,60,60
C
C  COMPUTE MATRIX ELEMENTS FOR V
C
60    TEMD=Q*SL*FPRZ(FPRY(XMMV(5),XMYV(5),XMPV(5)),FWY,
     1 FPRY(XPMV(5),XPYV(5),XPPV(5)))
      TEMQ=FPRY(T(8,3)*XZMV(1),T(8,1)*XV(1),T(8,2)*XZPV(1))
     1 +FPRZ(T(10,5)*XMYV(1),T(10,1)*XV(1),T(10,4)*XPYV(1))
      TEMC=FPRZ(XMYV(4)*XMYV(5),XV(4)*XV(5),XPYV(4)*XPYV(5))
      XMAT(4)=-TMU*SYM+ADM
      YMAT(4)=DXX-TMU*SY+AD+TEMK
      ZMAT(4)=-TMU*SYP+ADP
      DVEC(4)=DXX*XV(4)+TMU*SPRZ(XMYV(4),XV(4),XPYV(4))
     1 +TEMD-FPRY(TZMV(6),TV(6),TZPV(6))-TEMQ+TEMK*ZEROV(4)
     2 +Q*SL*SPRZ(XMYV(4),XV(4),XPYV(4))+SL*FQZ*(FWY+FVZ)-TEMC
C
C  COMPUTE MATRIX ELEMENTS FOR W
C
      TEMD=Q*SL*FPRY(FPRZ(XMMV(4),XZMV(4),XPMV(4)),FVZ,
     1 FPRZ(XMPV(4),XZPV(4),XPPV(4)))
      TEMQ=FPRY(T(10,3)*XZMV(1),T(10,1)*XV(1),T(10,2)*XZPV(1))
     1 +FPRZ(T(9,5)*XMYV(1),T(9,1)*XV(1),T(9,4)*XPYV(1))
      TEMC=FPRZ(XMYV(5)*XMYV(5),XV(5)*XV(5),XPYV(5)*XPYV(5))
      XMAT(5)=-TMU*SYM+ADM-SL*(Q*SYM+FQY*FYM)
      YMAT(5)=DXX-TMU*SY+AD+TEMK-SL*(Q*SY+FQY*FY)
      ZMAT(5)=-TMU*SYP+ADP-SL*(Q*SYP+FQY*FYP)
      DVEC(5)=DXX*XV(5)+TMU*SPRZ(XMYV(5),XV(5),XPYV(5))
     1 +TEMD-FPRZ(TMYV(6),TV(6),TPYV(6))-TEMQ+TEMK*ZEROV(5)
     2 +SL*FQY*FVZ-G*XV(2)-TEMC
C
C  COMPUTE MATRIX ELEMENTS FOR SCALE
C
70    IF (XV(1)-QCUT*FMAXV(1)) 80,80,90
80    DVEC(6)=2.0*SCALE
      GO TO 100
90    TEMD=SPLZ*SPRZ(XMYV(6),XV(6),XPYV(6))+FPLZ*FLZ
      TEMC=FPRZ(XMYV(6)*XMYV(5),XV(6)*XV(5),XPYV(6)*XPYV(5))
      XMAT(6)=XMAT(1)
      YMAT(6)=DXX-SPLY*SY-FPLY*FY+AD
     1 +S1*(PROM+PROP/XV(1))+S2*CL/2.0+S5*G*(T(3,1)+T(4,1)*FRZ/XV(1))
     2 +S7*XV(6)*(FQY*FWY+FQZ*FQZ)/Q+S8*(FQY*FLY+FQZ*FLZ)
      ZMAT(6)=ZMAT(1)
      DVEC(6)=DXX*XV(6)+TEMD-S6*Q*(FLY*FLY+FLZ*FLZ)-TEMC
C
C  LOWER BOUNDARY CONDITION CHECK
C
100   IF (JY-2) 200,110,105
105   IF (JY-IYPSN) 300,300,500
110   IF (LYLFF) 500,300,300
```

```
200   DO 220 I=1,NWVEC
      TEM=1.0
      IF (IRFV(I,1)) 205,205,210
205   TEM=-1.0
210   ZMAT(I)=ZMAT(I)+TEM*XMAT(I)
220   CONTINUE
      GO TO 400
300   DO 320 I=1,NWVEC
      DVEC(I)=DVEC(I)-XMAT(I)*ZEROV(I)
320   CONTINUE
400   CALL SFVFL(0.0,XMAT,NWVEC)
500   RETURN
      END
CART ID 0105  DB ADDR 45C0   DB CNT 017A
```

```
WAKMZ.S(0105)
**WAKMZ - STRATIFIED SUBMARINE WAKE, Z IMPLICIT MATRIX COEFFICIENTS
      SUBROUTINE WAKMZ
C
C  THIS SUBROUTINE IN THE WAKE PROGRAM COMPUTES THE X,Y,Z AND D MATRIX
C  COEFFICIENTS FOR THE Z IMPLICIT UPSWEEP
C
      DIMENSION T(10,5)
*COPY (CMWAK)
      EQUIVALENCE (T(1,1),ZNEWV(1))
C
      FPRZ(ARGM,ARG,ARGP)=FZM*(ARGM+FZS*ARG-FZR*ARGP)
      FPRY(ARGM,ARG,ARGP)=FYM*(ARGM+FYS*ARG-FYR*ARGP)
      SPRZ(ARGM,ARG,ARGP)=SZM*ARGM+SZ*ARG+SZP*ARGP
      SPRY(ARGM,ARG,ARGP)=SYM*ARGM+SY*ARG+SYP*ARGP
C
C  COMPUTE SPACING FACTORS
C
      DZM=Z-ZM
      DZP=ZP-Z
      DZT=DZM+DZP
      FZM=-DZP/DZM/DZT
      FZP=DZM/DZP/DZT
      FZ=-FZM-FZP
      TEM=DZM/DZP
      FZR=TEM*TEM
      FZS=FZR-1.0
      SZM=2.0/DZM/DZT
      SZP=2.0/DZP/DZT
      SZ=-SZM-SZP
C
      DYM=Y-YM
      DYP=YP-Y
      DYT=DYM+DYP
      FYM=-DYP/DYM/DYT
      FYP=DYM/DYP/DYT
      FY=-FYM-FYP
      TEM=DYM/DYP
      FYR=TEM*TEM
      FYS=FYR-1.0
      SYM=2.0/DYM/DYT
      SYP=2.0/DYP/DYT
      SY=-SYM-SYP
C
      CALL SFVFL(0.0,XMAT,NWVEC)
      CALL SFVFL(1.0,YMAT,NWVEC)
      CALL SFVFL(0.0,ZMAT,NMOVE)
C
C  COMPUTE MULTIPLICATIVE FACTORS
C
      CALL WAKDG(ZM)
      CALL WAKMV(XMYV,TMYV,T(1,5))
      CALL WAKDG(ZP)
      CALL WAKMV(XPYV,TPYV,T(1,4))
      CALL WAKDG(Z)
      CALL WAKMV(XZMV,TZMV,T(1,3))
      CALL WAKMV(XZPV,TZPV,T(1,2))
      CALL WAKMV(XV,TV,T(1,1))
      TMU=FMU
```

```
      IF (SPORS) 40,40,20
20    TEM=Y*Y/YSCAL/YSCAL+Z*Z/ZSCAL/ZSCAL
      IF (TEM-PCUT) 40,40,30
30    TMU=CMU
40    CL=2.0*(AS*TMU/SCALE+B*Q)/SCALE
C
C  COMPUTE DERIVATIVE FACTORS
C
      DXX=DXI*(U+XV(3))
      CALL WAKLL(TEMK,1)
      ADM=FZM*XMYV(5)
      AD=FZ*XV(5)
      ADP=FZP*XPYV(5)
      SL=SCALE/3.0
      FQY=FPRY(SQRT(XZMV(1)),Q,SQRT(XZPV(1)))
      FQZ=FPRZ(SQRT(XMYV(1)),Q,SQRT(XPYV(1)))
      FKY=FPRY(XZMV(1),XV(1),XZPV(1))
      FRY=FPRY(XZMV(2),XV(2),XZPV(2))
      FRZ=FPRZ(XMYV(2),XV(2),XPYV(2))
      FUY=FPRY(XZMV(3),XV(3),XZPV(3))
      FUZ=FPRZ(XMYV(3),XV(3),XPYV(3))
      FVY=FPRY(XZMV(4),XV(4),XZPV(4))
      FVZ=FPRZ(XMYV(4),XV(4),XPYV(4))
      FWY=FPRY(XZMV(5),XV(5),XZPV(5))
      FWZ=FPRZ(XMYV(5),XV(5),XPYV(5))
      FLY=FPRY(XZMV(6),XV(6),XZPV(6))
      FLZ=FPRZ(XMYV(6),XV(6),XPYV(6))
      SPLY=3.0*DC*YSCAL*CVS*Q+TMU
      FPLY=3.0*DC*YSCAL*CVS*FQY
      SPLZ=3.0*DC*SCAL*CWS*Q+TMU
      FPLZ=3.0*DC*SCAL*CWS*FQZ
      PROM=FUY*T(5,1)+FUZ*T(1,1)+FVY*T(8,1)+(FWY+FVZ)*T(10,1)+FWZ*T(9,1)
      PROP=FUY*FUY*T(6,1)+FUZ*FUZ*T(2,1)-Q*SCALE*(FWY+FVZ)**2/3.0
C
C  COMPUTE MATRIX ELEMENTS FOR QQ
C
      TEMD=SPLY*SPRY(XZMV(1),XV(1),XZPV(1))+FPLY*FKY
      TEMC=FPRY(XZMV(1)*XZMV(4),XV(1)*XV(4),XZPV(1)*XZPV(4))
      XMAT(1)=-SPLZ*SZM-FPLZ FZM+ADM
      YMAT(1)=DXX-SPLZ*SZ-FPLZ*FZ+AD+CL+2.0*(PROM+G*T(3,1))
      ZMAT(1)=-SPLZ*SZP-FPLZ*FZP+ADP
      DVEC(1)=DXX*XV(1)-2.0*(PROP+G*FRZ*T(4,1))+TEMD-TEMC
C
C  COMPUTE MATRIX ELEMENTS FOR RHO
C
      TEMM=T(4,1)-XK
      TEMP=FPRZ(T(4,5),T(4,1),T(4,4))
      TEMD=(T(7,1)-XK)*SPRY(XZMV(2),XV(2),XZPV(2))
     1 +FPRY(T(7,3),T(7,1),T(7,2))*FRY
      TEMQ=FPRZ(T(3,5)*XMYV(1),T(3,1)*XV(1),T(3,4)*XPYV(1))
      TEMC=FPRY(XZMV(2)*XZMV(4),XV(2)*XV(4),XZPV(2)*XZPV(4))
      XMAT(2)=TEMM*SZM+TEMP*FZM+ADM
      YMAT(2)=DXX+TEMM*SZ+TEMP*FZ+AD
      ZMAT(2)=TEMM*SZP+TEMP*FZP+ADP
      DVEC(2)=DXX*XV(2)-TEMD-TEMQ-DRDZ*XV(5)-TEMC
      IF (NSTAT-2) 50,60,50
C
C  COMPUTE MATRIX ELEMENTS FOR U
C
```

```
50    TEMM=T(2,1)-TMU
      TEMP=FPRZ(T(2,5),T(2,1),T(2,4))
      TEMD=(T(6,1)-TMU)*SPRY(XZMV(3),XV(3),XZPV(3))
     1 +FPRY(T(6,3),T(6,1),T(6,2))*FUY
      TEMQ=FPRY(T(5,3)*XZMV(1),T(5,1)*XV(1),T(5,2)*XZPV(1))
     1 +FPRZ(T(1,5)*XMYV(1),T(1,1)*XV(1),T(1,4)*XPYV(1))
      TEMC=FPRY(XZMV(3)*XZMV(4),XV(3)*XV(4),XZPV(3)*XZPV(4))
      XMAT(3)=TEMM*SZM+TEMP*FZM+ADM
      YMAT(3)=DXX+TEMM*SZ+TEMP*FZ+AD
      ZMAT(3)=TEMM*SZP+TEMP*FZP+ADP
      DVEC(3)=DXX*XV(3)-TEMD-TEMQ-TEMC
      IF (NSTAT-2) 70,60,60
C
C  COMPUTE MATRIX ELEMENTS FOR V
C
60    TEMD=Q*SL*FPRZ(FPRY(XMMV(5),XMYV(5),XMPV(5)),FWY,
     1 FPRY(XPMV(5),XPYV(5),XPPV(5)))
      TEMQ=FPRY(T(8,3)*XZMV(1),T(8,1)*XV(1),T(8,2)*XZPV(1))
     1 +FPRZ(T(10,5)*XMYV(1),T(10,1)*XV(1),T(10,4)*XPYV(1))
      TEMC=FPRY(XZMV(4)*XZMV(4),XV(4)*XV(4),XZPV(4)*XZPV(4))
      XMAT(4)=-TMU*SZM+ADM-SL*(Q*SZM+FQZ*FZM)
      YMAT(4)=DXX-TMU*SZ+AD+TEMK-SL*(Q*SZ+FQZ*FZ)
      ZMAT(4)=-TMU*SZP+ADP-SL*(Q*SZP+FQZ*FZP)
      DVEC(4)=DXX*XV(4)+TMU*SPRY(XZMV(4),XV(4),XZPV(4))
     1 +TEMD-FPRY(TZMV(6),TV(6),TZPV(6))-TEMQ+TEMK*ZEROV(4)
     2 +SL*FQZ*FWY-TEMC
C
C  COMPUTE MATRIX ELEMENTS FOR W
C
      TEMD=Q*SL*FPRY(FPRZ(XMMV(4),XZMV(4),XPMV(4)),FVZ,
     1 FPRZ(XMPV(4),XZPV(4),XPPV(4)))
      TEMQ=FPRY(T(10,3)*XZMV(1),T(10,1)*XV(1),T(10,2)*XZPV(1))
     1 +FPRZ(T(9,5)*XMYV(1),T(9,1)*XV(1),T(9,4)*XPYV(1))
      TEMC=FPRY(XZMV(5)*XZMV(4),XV(5)*XV(4),XZPV(5)*XZPV(4))
      XMAT(5)=-TMU*SZM+ADM
      YMAT(5)=DXX-TMU*SZ+AD+TEMK
      ZMAT(5)=-TMU*SZP+ADP
      DVEC(5)=DXX*XV(5)+TMU*SPRY(XZMV(5),XV(5),XZPV(5))
     1 +TEMD-FPRZ(TMYV(6),TV(6),TPYV(6))-TEMQ+TEMK*ZEROV(5)
     2 +Q*SL*SPRY(XZMV(5),XV(5),XZPV(5))+SL*FQY*(FWY+FVZ)-G*XV(2)-TEMC
C
C  COMPUTE MATRIX ELEMENTS FOR SCALE
C
70    IF (XV(1)-QCUT*FMAXV(1)) 80,80,90
80    DVEC(6)=2.0*SCALE
      GO TO 100
90    TEMD=SPLY*SPRY(XZMV(6),XV(6),XZPV(6))+FPLY*FLY
      TEMC=FPRY(XZMV(6)*XZMV(4),XV(6)*XV(4),XZPV(6)*XZPV(4))
      XMAT(6)=XMAT(1)
      YMAT(6)=DXX-SPLZ*SZ-FPLZ*FZ+AD
     1 +S1*(PROM+PROP/XV(1))+S2*CL/2.0+S5*G*(T(3,1)+T(4,1)*FRZ/XV(1))
     2 +S7*XV(6)*(FQY*FWY+FQZ*FQZ)/Q+S8*(FQY*FLY+FQZ*FLZ)
      ZMAT(6)=ZMAT(1)
      DVEC(6)=DXX*XV(6)+TEMD-S6*Q*(FLY*FLY+FLZ*FLZ)-TEMC
C
C  LOWER BOUNDARY CONDITION CHECK
C
100   IF (JZ-1) 200,110,102
102   IF (JY-IYPSM) 104,500,106
```

```
104   IF (JY-1) 500,500,300
106   IF (JY-IYPEM) 500,500,300
110   IF (LZLFF) 500,300,300
200   DO 220 I=1,NWVEC
      TEM=1.0
      IF (IRFV(I,2)) 205,205,210
205   TEM=-1.0
210   ZMAT(I)=ZMAT(I)+TEM*XMAT(I)
220   CONTINUE
      GO TO 400
300   DO 320 I=1,NWVEC
      DVEC(I)=DVEC(I)-XMAT(I)*ZEROV(I)
320   CONTINUE
400   CALL SFVFL(0.0,XMAT,NWVEC)
500   RETURN
      END
CART ID 0105  DB ADDR 4740   DB CNT 0180
```

```
WAKOT,S(C105)
**WAKOT - STRATIFIED SUBMARINE WAKE, OUTPUT THE RESULTS
      SUBROUTINE WAKOT
C
C  THIS SUBROUTINE IN THE WAKE PROGRAM OUTPUTS THE RESULTS
C
      DIMENSION JVCHV(7),JPOS(2),PBUF(880),KLOCK(6),HDSV(2,2)
      DIMENSION TOTAL(7,40,2),OVEC(33),DBUF(440),NN(40),MM(40),HNUL(2)
*COPY (CMWAK)
      EQUIVALENCE (TOTAL(1,1,1),ROWG(1,1))
      EQUIVALENCE (PBUF(1),ROWB(1,1,3)),(PBUF(441),DBUF(1))
      EQUIVALENCE (NN(1),NRNZV(1)),(MM(1),NRNYV(1))
C
      DATA JPOS/1HY,1HZ/,JERRX/2H  /,HNUL/4H    ,4HNULL/
      DATA JVCHV/2HQQ,2HRO,2HU ,2HV ,2HW ,2HSL,2HP /
      DATA HDSV/4H    ,4H    ,4H(STO,4HRED)/
C
1000  FORMAT(/6X,24HFULL PROFILE OUTPUT FOR ,A2,1X,2A4,1X,A4)
1001  FORMAT(1H )
1003  FORMAT(/7X,5HZ   Y,10E12.4)
1004  FORMAT(/6X,23HMESH PROFILE OUTPUT AT ,A1,2H = ,E12.4//
     1 7X,A1,7(10X,A2))
1010  FORMAT(/6X,32HTURBULENCE PROFILE OUTPUT AT Z = ,E12.4//
     1 7X,1HY,10X,2HUV,10X,2HUW,10X,2HVW,10X,2HUR,10X,2HVR,10X,2HWR,
     2 10X,2HRR,10X,2HUU,10X,2HVV,10X,2HWW/)
1015  FORMAT(/6X,8HX POINTS,9X,1HX,13X,2HDX,10X,
     1 8HY SPREAD,7X,8HY POINTS,7X,8HZ SPREAD,7X,
     2 8HZ POINTS,9X,3HX/D,11X,3HB V/I10,4X,3E15.5,
     3 I9,6X,E15.5,I9,6X,2E14.5)
1021  FORMAT(//6X,6HSTATUS,7X,8HZ LAMBDA,7X,8HR LAMBDA,
     1 7X,8HY LAMBDA,7X,8HMOMENTUM,
     2 7X,8HP ENERGY,7X,8HK ENERGY,14X,12HELAPSED TIME/
     3 6X,A4,4X,6E15.5,8X,I2,5H HRS ,I2,5H MIN ,I2,4H SEC)
1036  FORMAT(/14X,7(10X,A2))
1037  FORMAT(18H MAXIMUM VALUE    ,7E12.4)
1038  FORMAT(18H Y LOCATION       ,7E12.4)
1039  FORMAT(18H Z LOCATION       ,7E12.4)
1040  FORMAT(18H MAXIMUM CHANGE   ,7E12.4)
1041  FORMAT(18H GLOBAL MAXIMUM   ,7E12.4)
1045  FORMAT(/18X,2HUV,10X,2HUW,10X,2HVW,10X,2HUR,10X,2HVR,10X,2HWR,
     1 10X,2HRR,10X,2HUU,10X,2HVV,10X,2HWW/12H MAXIMUMS   ,10E12.4)
1049  FORMAT(//6X,6HSTATUS,8X,8HX POINTS,9X,1HX/6X,A4,I14,E19.5)
2000  FORMAT(//19H BACKUP NEEDED FOR ,A2)
2001  FORMAT(//26H DIVIDE CHECK HAS OCCURRED)
2002  FORMAT(//22H OVERFLOW HAS OCCURRED)
2003  FORMAT(//33H SOLUTION HAS SPREAD TOO FAST IN ,A1,5X,4I6)
2005  FORMAT(//22H RUN SUSPENDED AT X = ,E12.5,5X,11HX POINTS = ,I3,
     1 5X,6HNGS = ,I3,5X,6HNPS = ,I3)
2006  FORMAT(//48H MAXIMUM POINTS EXCEEDED ON AUTOPOINT ADJUST IN ,A1)
2007  FORMAT(//22H GLOBAL FILE IS FILLED)
2008  FORMAT(//20H PLOT FILE IS FILLED)
2009  FORMAT(//4H WAK,A2,19H ERROR HAS OCCURRED)
2010  FORMAT(/20H TOTAL ELAPSED TIME ,I2,5H HRS ,I2,5H MIN ,I2,4H SEC)
2011  FORMAT(//49H TOTAL PROFILES STORED AT THIS X ON THE PLOT FILE)
2012  FORMAT(//26H MAXIMUM RUN TIME EXCEEDED)
2031  FORMAT(11E12.4)
2032  FORMAT(E12.4,12X,9E12.4)
2033  FORMAT(E12.4,24X,8E12.4)
2034  FORMAT(E12.4,36X,7E12.4)
```

```
2035  FORMAT(E12.4,48X,6E12.4)
2036  FORMAT(E12.4,60X,5E12.4)
2037  FORMAT(E12.4,72X,4E12.4)
2038  FORMAT(E12.4,84X,3E12.4)
2039  FORMAT(E12.4,96X,2E12.4)
2040  FORMAT(E12.4,108X,E12.4)
C
C  CHECK FOR ERRORS
C
      IOLAY=10
      MOOD=1
      NRSTX=NRST-1
      NREND=NRST+JEND
      NZP=JEND+2
      NYPSX=NYPS-1
      NYPEX=NYPE+1
      NYP=NYPE-NYPS+3
      LTRNF=LTRNF+LDVCF+LOVFF+LPKRF+LFCUR
      IF (LBURF) 105,105,100
100   WRITE (NOUT,2000) JVCHV(LBURF)
105   IF (JERR-JERRX) 106,107,106
106   WRITE (NOUT,2009) JERR
      LTRNF=1
107   IF (LTRNF) 110,200,110
110   IF (LDVCF) 120,130,120
120   WRITE (NOUT,2001)
130   IF (LOVFF) 140,150,140
140   WRITE (NOUT,2002)
150   IF (LPKRF) 160,190,160
160   WRITE (NOUT,2003) JPOS(LPKRF),NYPS,NYPE,NRST,JEND
190   IF (LFCUR) 195,200,195
195   WRITE (NOUT,2006) JPOS(LFCUR)
200   IF (LPRFL) 205,900,900
C
C  CHECK FOR INTERMEDIATE PRINTOUT TO LINE PRINTER
C
205   IF (LIOLF) 210,300,210
210   CALL CLOCK(1,KLOCK)
      Y=YOLDV(NYPEX)-YOLDV(NYPSX)
      Z=ZOLDV(NREND)-ZOLDV(NRSTX)
      YP=XFACT*XP+XZERO
      ZP=YP*SQRT(G)/3.1416
      WRITE (NOUT,1021) JSTAT,(EPSSV(I),I=1,3),XMOM,XPE,XKE,
     1 (KLOCK(I),I=1,6,2)
      WRITE (NOUT,1015) NPTSN,XP,DX,Y,NYP,Z,NZP,YP,ZP
      WRITE (NOUT,1045) TURBX
      WRITE (NOUT,1036) JVCHV
      WRITE (NOUT,1001)
      WRITE (NOUT,1037) FMAXV
      WRITE (NOUT,1038) YMAXV
      WRITE (NOUT,1039) ZMAXV
      WRITE (NOUT,1040) TMAXV
      WRITE (NOUT,1041) GMAXV
C
C  CHECK FOR INTERMEDIATE PRINTOUT TO GLOBAL DISK FILE
C
300   IF (LIOPF) 310,400,310
310   NRX=33
      IF (NCFRL(GLOID,NG,NRX+1)) 320,330,330
```

```
320   LTRNF=1
      WRITE (NOUT,2007)
      GO TO 400
330   OVEC(1)=XP
      CALL SFVMV(YSCAL,OVEC(2),2)
      CALL SFVMV(XMOM,OVEC(4),20)
      CALL SFVMV(GM,OVEC(24),8)
      OVEC(32)=GM(11)
      OVEC(33)=DEPST
      CALL PBFDW(GLOID,NG,NRX,OVEC)
      NRX=NG
      CALL PBFDW(GLOID,NRX,1,-1,0)
C
C  CHECK FOR MESH PRINTOUT TO LINE PRINTER
C
400   IF (LFOLF) 404,450,404
404   WRITE (NOUT,1049) JSTAT,NPTSN,XP
      DO 409 I=1,NVART
      IF (FMAXV(I)) 409,409,405
405   DO 406 JY=NYPSX,NYPEX
      TOTAL(I,JY,1)=TOTAL(I,JY,1)/FMAXV(I)
406   CONTINUE
      DO 407 JZ=NRSTX,NREND
      TOTAL(I,JZ,2)=TOTAL(I,JZ,2)/FMAXV(I)
407   CONTINUE
409   CONTINUE
      ZM=EPSSV(1)/2.0/C
      YM=EPSSV(3)/2.0/C
      WRITE (NOUT,1004) JPOS(2),ZOUT,JPOS(1),JVCHV
      WRITE (NOUT,1001)
      DO 410 JY=NYPSX,NYPEX
      Y=YOLDV(JY)/YM
      WRITE (NOUT,2031) Y,(TOTAL(I,JY,1),I=1,NVART)
410   CONTINUE
      WRITE (NOUT,1004) JPOS(1),YOUT,JPOS(2),JVCHV
      WRITE (NOUT,1001)
      DO 420 JZ=NRSTX,NREND
      Z=ZOLDV(JZ)/ZM
      WRITE (NOUT,2031) Z,(TOTAL(I,JZ,2),I=1,NVART)
420   CONTINUE
C
C  CHECK FOR TOTAL TURBULENCE PRINTOUT TO LINE PRINTER
C
450   IF (LSOLF) 460,500,460
460   IF (LAMIN) 500,461,500
461   WRITE (NOUT,1049) JSTAT,NPTSN,XP
      GMAX=1.0E-06*FMAXV(1)
      DO 480 JZ=NRSTX,NREND
      IYPS=IYPSV(JZ)
      IYPE=IYPEV(JZ)
      IF (IYPS) 480,480,462
462   IYPSX=IYPS+LYLFF
      CALL WAKMG(JZ,ZV,1)
      IYS=0
      DO 468 JY=IYPSX,IYPE
      DO 465 I=1,10
      IF (ABS(ROWG(I,JY))-GMAX) 465,465,466
465   CONTINUE
      GO TO 468
```

```
466   IYE=JY
      IF (IYS) 468,467,468
467   IYS=JY
468   CONTINUE
      IF (IYS) 469,480,469
469   WRITE (NOUT,1010) ZOLDV(JZ)
      DO 470 JY=IYS,IYE
      WRITE (NOUT,2031) YOLDV(JY),(ROWG(I,JY),I=1,10)
470   CONTINUE
480   CONTINUE
C
C  CHECK FOR TOTAL PRINTOUT TO PROFILE DISK FILE
C
500   IF (LTOPF) 510,600,510
510   IVECA=NYP
      IVECB=NZP
      J=0
      DO 515 I=1,NVART
      J=J+IDSV(I)
515   CONTINUE
      NRX=J*IVECA*IVECB+IVECA+IVECB+3
      IF (NDFRL(PLTID,NP,NRX)) 520,530,530
520   LTRNF=1
      LTOPF=0
      WRITE (NOUT,2008)
      GO TO 600
530   ZA=XP
      CALL PBFDW(PLTID,NP,2,ZA)
      CALL PBFDW(PLTID,NP,IVECA,YOLDV(NYPSX))
      CALL PBFDW(PLTID,NP,IVECB,ZOLDV(NRSTX))
      WRITE (NOUT,2011)
      GO TO 601
C
C  CHECK FOR TOTAL PRINTOUT TO LINE PRINTER
C
600   IF (LTOLF) 601,700,601
601   IROWR=2
      IF (LBURF) 604,604,603
603   IROWR=4
604   IF (LTOLF) 608,610,608
608   WRITE (NOUT,1049) JSTAT,NPTSN,XP
610   DO 680 IVAR=1,NVART
      CALL SFVFL(ZEROV(IVAR),PBUF,880)
      K=1
      GMAX=GMAXV(IVAR)
      JVAR=IVAR
      IF (IVAR-NVART) 612,611,611
611   JVAR=6
      IROWR=IROWT
612   GO TO (6125,6125,6121,6122,6122,6125,6122),IVAR
6121  IF (NSTAT-2) 6125,680,6125
6122  IF (NSTAT-1) 6125,680,6125
6125  IF (LTOLF) 613,614,613
613   IF (GMAX) 6132,6132,6135
6132  K=2
6135  I=MMIN(1,LTOPF)*IDSV(IVAR)+1
      WRITE (NOUT,1000) JVCHV(IVAR),(HDSV(J,I),J=1,2),HNUL(K)
614   DO 670 IYP=1,NYP,10
      IYPSX=NYPSX+IYP-1
```

```
      IYPEX=MMIN(IYPSX+9,NYPEX)
      N=IYPEX-IYPSX+1
      K=N+1
      NZP=440/K*K
      IF (LTOLF) 616,618,616
616   IF (GMAX) 618,618,6165
6165  WRITE (NOUT,1003) (YOLDV(I),I=IYPSX,IYPEX)
      WRITE (NOUT,1001)
618   I=0
      J=0
      L=0
      DO 650 NR=NRSTX,NREND
      NPOS=NR
      I=I+1
      CALL WAKRR(NR,ZV)
      PBUF(I)=Z
      IF (NR-NRST) 6181,6183,6184
6181  IF (LZLFF) 6187,6182,6182
6182  IYPS=IYPSV(NR+1)
      IYPE=IYPEV(NR+1)
      GO TO 619
6183  IF (LZLFF) 6186,6187,6187
6184  IF (NR-NREND) 6186,6185,6185
6185  IYPS=IYPSV(NR-1)
      IYPE=IYPEV(NR-1)
      GO TO 619
6186  IYPS=MMIN(IYPS,IYPSV(NR-1))
      IYPE=MMAX(IYPE,IYPEV(NR-1))
      IF (NR-NREND+1) 6187,619,619
6187  IYPS=MMIN(IYPS,IYPSV(NR+1))
      IYPE=MMAX(IYPE,IYPEV(NR+1))
619   L=L+1
      NN(L)=0
      MM(L)=1
      DO 640 JY=IYPSX,IYPEX
      I=I+1
      J=J+1
      IF (JY-IYPS+1) 6195,630,620
6195  MM(L)=MM(L)+1
      GO TO 640
620   IF (JY-IYPE-1) 630,630,621
621   I=I+IYPEX-JY
      J=J+IYPEX-JY
      GO TO 641
630   CALL WAKMP(IROWR,NR,JY,XV,1)
      PBUF(I)=XV(JVAR)
      DBUF(J)=XV(JVAR)
      NN(L)=NN(L)+1
640   CONTINUE
641   IF (I-NZP) 6415,642,642
6415  IF (NR-NREND) 650,642,642
642   I=I/K
      IF (LTOPF) 6432,6438,6432
6432  IF (IDSV(IVAR)) 6434,6436,6434
6434  CALL PBFDW(PLTID,NP,J,DBUF)
6436  IF (LTOLF) 6438,646,6438
6438  IF (GMAX) 646,646,644
644   DO 645 JZ=1,I
      IF (NN(JZ)) 645,645,6440
```

```
6440  J=MM(JZ)
      L=(JZ-1)*K+1
      JS=L+J
      JE=JS+NN(JZ)-1
      GO TO (6441,6442,6443,6444,6445,6446,6447,6448,6449,6450),J
6441  WRITE (NOUT,2031) PBUF(L),(PBUF(J),J=JS,JE)
      GO TO 645
6442  WRITE (NOUT,2032) PBUF(L),(PBUF(J),J=JS,JE)
      GO TO 645
6443  WRITE (NOUT,2033) PBUF(L),(PBUF(J),J=JS,JE)
      GO TO 645
6444  WRITE (NOUT,2034) PBUF(L),(PBUF(J),J=JS,JE)
      GO TO 645
6445  WRITE (NOUT,2035) PBUF(L),(PBUF(J),J=JS,JE)
      GO TO 645
6446  WRITE (NOUT,2036) PBUF(L),(PBUF(J),J=JS,JE)
      GO TO 645
6447  WRITE (NOUT,2037) PBUF(L),(PBUF(J),J=JS,JE)
      GO TO 645
6448  WRITE (NOUT,2038) PBUF(L),(PBUF(J),J=JS,JE)
      GO TO 645
6449  WRITE (NOUT,2039) PBUF(L),(PBUF(J),J=JS,JE)
      GO TO 645
6450  WRITE (NOUT,2040) PBUF(L),(PBUF(J),J=JS,JE)
645   CONTINUE
646   I=0
      J=0
      L=0
      CALL SFVFL(ZEROV(IVAR),PBUF,880)
650   CONTINUE
670   CONTINUE
680   CONTINUE
      IF (LTOPF) 690,700,690
690   NRX=NP
      CALL PBFDW(PLTID,NRX,1,-1.0)
C
C  WRITE COMPLETED SOLUTION FOR NEXT INTEGRATION STEP (WAKSC)
C
700   IF (LMLFL) 702,800,800
702   NPTS=NPTSN
      X=XP
      IF (LSTFL) 705,710,705
705   LSTFL=0
710   NRX=1
      CALL PBFDW(COMID,NRX,NCOMT,NSTST)
C
C  SET UP FOR RETURN OR END
C
800   IF (LTRNF) 900,850,900
850   N=1
      GO TO 910
860   IF (KLOCK(1)-MXHRS) 870,880,880
870   IF (JOBE) 875,875,872
872   IF (JS-JOBE) 875,900,900
875   RETURN
880   WRITE (NOUT,2012)
900   N=2
      JS=NG/160+MMIN(1,NIOPP)
      JE=NP/160+MMIN(1,NTOPP)
```

```
      WRITE (NOUT,2005) XP,NPTSN,JS,JE
910   CALL CLOCK(1,KLOCK)
      JS=KLOCK(3)
      DO 924 I=1,6,2
      J=6-I
      K=J-2
      JE=KLOK(J)+KLOCK(J)
      IF (K) 923,923,921
921   IF (JE-59) 923,923,922
922   JE=JE-60
      KLOCK(K)=KLOCK(K)+1
923   KLOCK(J)=JE
924   CONTINUE
      GO TO (860,930),N
930   DO 932 I=1,6,2
      KLOK(I)=KLOCK(I)
932   CONTINUE
      WRITE (NOUT,2010) (KLOK(I),I=1,6,2)
      NRX=1
      CALL PBFDW(COMID,NRX,NCOMT,NSTST)
      CALL CLBFR(BUFR)
      CALL CLBFR(BUFS)
      CALL EXIT
      END
CART ID 0105  DB ADDR 5270   DB CNT 02F2
```

```
WAKPC.S(0105)
**WAKPC - STRATIFIED SUBMARINE WAKE, PRESSURE CALCULATION
      SUBROUTINE WAKPC
C
C  THIS SUBROUTINE IN THE WAKE PROGRAM CONTROLS THE SOLUTION
C  FOR THE PRESSURE BY THE POISSON EQUATION
C
      DIMENSION TOTAL(7,40,2),PN(5),VALUE(40,2)
*COPY (CMWAK)
      EQUIVALENCE (TOTAL(1,1,1),ROWG(1,1)),(MM,EPSXV(1))
      EQUIVALENCE (ERROR,GM(12)),(GM(10),DUI),(GM(9),DLI)
      EQUIVALENCE (VALUE(1,1),ROWG(1,36))
C
      DATA PN/10.,30.,60.,100.,150./
C
1000  FORMAT(//6X,6HSTATUS,7X,9HXX POINTS,6X,9HMAX VALUE,6X,
     1 10HMAX CHANGE,8X,3HDXX,11X,5HERROR/8X,2HPR,I14,5X,4E15.5)
2010  FORMAT(//41H PROGRAM EXIT DURING PRESSURE COMPUTATION)
C
C  CHECK RUN POSITION
C
      IOLAY=8
      IF (LSTFL) 50,105,100
50    LSTFL=LPRFL
      IF (LSTFL) 310,85,85
85    MPTS=MM
      DXX=EPSXV(2)
      GO TO 109
100   IF (LPRFL) 310,105,105
105   MPTS=0
      NREND=NRST+JEND
      Y=YOLDV(NYPE+1)-YOLDV(NYPS-1)
      Z=ZOLDV(NREND)-ZOLDV(NRST-1)
      DXX=PNORM*Y*Z/PN(1)
109   LPRFL=-1
      MOOD=0
      IBOT=1
      ITOP=MXRY
      LYFAF=0
      CALL SFVMV(ZEROV,XPPV,NWVEC)
      CALL SFVFL(0.0,ZEROV,NWVEC)
      IF (LFOLF) 1090,110,1090
1090  NYPSX=NYPS+LYLFF
      DO 1094 JY=NYPSX,NYPE
      IF (YOUT-YOLDV(JY)) 110,1096,1092
1092  IF (YOUT-YOLDV(JY+1)) 1096,1094,1094
1094  CONTINUE
1096  LYFAF=JY
      RATY=(YOUT-YOLDV(JY))/(YOLDV(JY+1)-YOLDV(JY))
      CALL SFVFL(0.0,VALUE,80)
C
C  INITIALIZE FOR NEXT INTEGRATION STEP
C
110   CALL DATSW(0,ISW)
      GO TO (111,114),ISW
111   WRITE (NOUT,2010)
      MM=MPTS
      EPSXV(2)=DXX
      LPRFL=LSTFL
```

```
      X=XP
      NPTS=NPTSN
      GO TO 300
114   DXI=2.0/DXX
      FMAXV(7)=0.0
      TMAXV(7)=0.0
      DUI=0.0
      DLI=0.0
      MPTS=MPTS+1
      CALL WAKPZ
C
C  DETERMINE NEW DXX FOR PRESSURE SOLUTION
C
      DXXO=DXX
      IF (MPTS-NSTPR) 127,170,170
127   DXX=DXX*PN(MPTS)/PN(MPTS+1)
C
C  SUCCESSFUL INTEGRATION STEP
C
170   ERROR=SQRT(DUI/DLI)/DXXO
      WRITE (NOUT,1000) MPTS,FMAXV(7),TMAXV(7),DXXO,ERROR
      IF (MPTS-1) 172,172,171
171   IF (TMAXV(7)-TEM) 172,172,180
172   TEM=TMAXV(7)
      IF (MPTS-NSTPR) 175,180,180
175   IF (TMAXV(7)/FMAXV(7)-PCRIT) 180,180,110
180   IF (LFOLF) 210,290,210
210   NRSTX=NRST+LZLFF
      NREND=NRST+JEND-1
      DO 230 NR=NRSTX,NREND
      IF (ZOUT-ZOLDV(NR)) 265,240,220
220   IF (ZOUT-ZOLDV(NR+1)) 240,230,230
230   CONTINUE
240   RATZ=(ZOUT-ZOLDV(NR))/(ZOLDV(NR+1)-ZOLDV(NR))
      NPOS=NR
      CALL WAKRR(NR,ZV)
      CALL WAKRR(NR+1,ZPV)
      IYPSX=IYPS+LYLFF
      DO 260 JY=IYPSX,IYPE
      CALL WAKMP(IROWT,NR,JY,XV,1)
      CALL WAKMP(IROWT,NR+1,JY,XPYV,1)
      TOTAL(7,JY,1)=XV(6)+RATZ*(XPYV(6)-XV(6))
260   CONTINUE
265   IF (LYFAF) 290,290,270
270   DO 280 NR=NRSTX,NREND
      TOTAL(7,NR,2)=VALUE(NR,1)+RATY*(VALUE(NR,2)-VALUE(NR,1))
280   CONTINUE
290   GMAXV(7)=FMAXV(7)
300   CALL SFVMV(XPPV,ZEROV,NWVEC)
310   RETURN
      END
CART ID 0105  DB ADDR 5A00   DB CNT 000A
```

```
WAKPI.S(0105)
**WAKPI - STRATIFIED SUBMARINE WAKE, PROFILE INITIALIZATION
      SUBROUTINE WAKPI
C
C  THIS SUBROUTINE IN THE WAKE PROGRAM INITIALIZES
C  THE WORKING FILE WITH
C
C  1) KNOWN JET PLOT FILE RESULTS FOR AXISYMMETRIC FLOW
C
C  2) KNOWN STRATIFIED WAKE PLOT FILE RESULTS FOR SPECIAL RESTART
C
C  3) GIVEN PRESSURE DISTRIBUTIONS FOR POISSONS EQUATION
C
      DIMENSION FILID(3),RBUF(6,40),RR(16),JTOP(2)
*COPY (CMWAK)
      EQUIVALENCE (RBUF(1,1),ROWG(1,1)),(AV(1),FILID(1))
      EQUIVALENCE (JTOP(2),ZNEWV(2)),(RR(1),GM(1))
C
1000  FORMAT(//24H IMPROPER INITIALIZATION,4I5)
C
      IOLAY=4
      CALL SFVFL(0.0,GMAXV,NVART)
      MOOD=-1
      IF (LYFAF) 210,200,50
C
C  FIND APPROPRIATE JET PROFILES TO INITIALIZE NSTAT = 1
C
50    NR=2
      NVARI=3
100   NP=NR-1
      CALL PBFDR(FILID,NP,LZFAF,ZNEWV)
      ITOP=JTOP(1)
      NR=NP+ITOP*LYFAF
      CALL PBFDR(FILID,NR,1,XP)
      IF (XP) 103,102,102
102   IF (XP-X) 100,100,103
103   NR=1
      X=ZNEWV(1)
      CALL SFVFL(0.0,XV,NVAR)
      XV(6)=SCALE
      IBOT=MMIN(MXRY/(LYLFF+2),MXRZ/(LZLFF+2))
      NSK=ITOP/IBOT+1
      JO=0
      NY=0
      DO 120 J=1,ITOP,NSK
      NY=NY+1
      NP=NP+(J-JO-1)*LYFAF
      JO=J
      CALL PBFDR(FILID,NP,LYFAF,RR)
      RBUF(1,NY)=RR(5)+RR(6)+RR(7)
      RBUF(2,NY)=0.0
      RBUF(3,NY)=RR(3)
      RBUF(4,NY)=RR(1)
      DO 115 I=1,NVARI
      TEM=ABS(RBUF(I,NY))
      IF (TEM-GMAXV(1)) 115,115,114
114   GMAXV(I)=TEM
115   CONTINUE
      IF (J-ITOP) 116,120,120
```

```
116   IF (J+NSK-ITOP) 120,120,118
118   J=ITOP-NSK
120   CONTINUE
      CALL WAKSE(EPSS,EPSSV)
122   CALL SFVFL(0.0,RBUF(1,NY),NVARI)
      DO 125 I=1,NVARI
      IF (ABS(RBUF(I,NY-1))-EPSSV(I)) 125,125,126
125   CONTINUE
      NY=NY-1
      GO TO 122
126   IF (LZLFF) 1262,1264,1264
1262  NRST=2
      NRSTX=1
      JEND=NY-2
      GO TO 1266
1264  NRST=(MXRZ+1)/2-NY+2
      NRSTX=MXRZ/2
      JEND=NY+NY-3
1266  IF (LYLFF) 1268,1270,1270
1268  NYPS=2
      NYPSX=1
      NYPE=NY-1
      GO TO 127
1270  NYPS=(MXRY+1)/2-NY+2
      NYPSX=MXRY/2
      NYPE=NYPS+NY+NY-4
127   IF (JEND) 129,129,1275
1275  IF (NRST-2) 129,128,128
128   IF (NYPS-NYPE) 1285,129,129
1285  IF (NYPS-2) 129,130,130
129   WRITE (NOUT,1000) NYPS,NYPE,NRST,JEND
      CALL EXIT
C
C  CONSTRUCT AXISYMMETRIC WORKING FILE
C
130   DO 180 JZ=1,NY
      NR=NRSTX+JZ-1
      NPOS=NR
      Z=RBUF(4,JZ)
      IF (JZ-NY) 132,131,131
131   IYPS=0
      IYPE=0
      GO TO 170
132   J=JZ
      IYPS=NYPSX-LYLFF
      DO 160 JY=1,NY
      J=J-1
      NRY=NYPSX+JY-1
      Y=RBUF(4,JY)
      IF (JZ-1) 133,133,135
133   YOLDV(NRY)=Y
      IF (LYLFF) 135,134,134
134   IY=NYPSX-JY+1
      YOLDV(IY)=-Y
135   R=SQRT(Y*Y+Z*Z)
140   J=J+1
      IF (JY-NY) 141,165,165
141   IF (R-RBUF(4,J+1)) 142,142,140
142   RATR=(R-RBUF(4,J))/(RBUF(4,J+1)-RBUF(4,J))
```

```
      CALL SFVMV(RBUF(1,J),XZMV,NVARI)
      CALL SFVMV(RBUF(1,J+1),XZPV,NVARI)
      IVAR=0
      DO 146 I=1,NVARI
      XV(I)=XZMV(I)+RATR*(XZPV(I)-XZMV(I))
      IF (ABS(XV(I))-EPSSV(I)) 145,145,146
145   IVAR=IVAR+1
146   CONTINUE
      IF (IVAR-NVARI) 150,165,165
150   CALL WAKMP(IROWR,NR,NRY,XV,2)
      IF (LYLFF) 160,151,151
151   IF (JY-1) 160,160,152
152   IY=NYPSX-JY+1
      CALL WAKMP(IROWR,NR,IY,XV,2)
      IYPS=IYPS-1
160   CONTINUE
165   IYPE=NRY-1
170   CALL WAKWR(NR,ZV)
      IF (LZLFF) 180,171,171
171   IF (JZ-1) 180,180,172
172   NR=NRSTX-JZ+1
      NPOS=NR
      Z=-Z
      CALL WAKWR(NR,ZV)
180   CONTINUE
      GO TO 250
C
C  FIND APPROPRIATE SPECIAL RESTART PROFILES
C
200   CALL WAKSR(FILID)
      GO TO 250
210   CALL WAKSR(PLTID)
250   NP=1
      NG=1
      RETURN
      END
CART ID 0105  DB ADDR 3AC0   DB CNT 0132
```

```
WAKPM.S(0105)
**WAKPM - STRATIFIED SUBMARINE WAKE, PRESSURE MANIPULATIONS
      SUBROUTINE WAKPM(LFL)
C
C  THIS SUBROUTINE IN THE WAKE PROGRAM PERFORMS VALUE MANIPULATIONS
C  ON THE UPSWEEP AND DOWNSWEEP OF THE PRESSURE IMPLICIT SOLUTION
C
      DIMENSION VALUE(40,2)
*COPY (CMWAK)
      EQUIVALENCE (GM(10),DUI),(GM(9),DLI)
      EQUIVALENCE (VALUE(1,1),ROWG(1,36))
C
      SPRZ(ARGM,ARG,ARGP)=SZM*ARGM+SZ*ARG+SZP*ARGP
      SPRY(ARGM,ARG,ARGP)=SYM*ARGM+SY*ARG+SYP*ARGP
C
      GO TO (100,200),LFL
C
C  MATRIX UPSWEEP CALCULATIONS
C
100   DZM=Z-ZM
      DZP=ZP-Z
      DZT=DZM+DZP
      SZM=2.0/DZM/DZT
      SZP=2.0/DZP/DZT
      SZ=-SZM-SZP
      DYM=Y-YM
      DYP=YP-Y
      DYT=DYM+DYP
      SYM=2.0/DYM/DYT
      SYP=2.0/DYP/DYT
      SY=-SYM-SYP
      GO TO (110,120),LZFAF
C
110   XMAT(1)=-SYM
      YMAT(1)=DXI-SY
      ZMAT(1)=-SYP
      DVEC(1)=DXI*XV(6)+SPRZ(XMYV(6),XV(6),XPYV(6))-TV(5)
      IF (JY-2) 1105,112,111
1105  TEM=FLOAT(IRFV(6,3))
      GO TO 122
111   IF (JY-IYPSN) 123,123,130
112   IF (LYLFF) 130,123,123
C
120   XMAT(1)=-SZM
      YMAT(1)=DXI-SZ
      ZMAT(1)=-SZP
      DVEC(1)=DXI*XV(6)+SPRY(XZMV(6),XV(6),XZPV(6))-TV(5)
      IF (JZ-1) 1205,1215,1208
1205  TEM=FLOAT(IRFV(6,4))
      GO TO 122
1208  IF (JY-IYPSM) 1209,130,121
1209  IF (JY-1) 130,130,123
121   IF (JY-IYPEM) 130,130,123
1215  IF (LZLFF) 130,123,123
C
122   ZMAT(1)=ZMAT(1)+(2.0*TEM-1.0)*XMAT(1)
123   XMAT(1)=0.0
C
130   TEM=1.0/(YMAT(1)-XMAT(1)*AV(1))
```

```
      AV(1)=TEM*ZMAT(1)
      AV(6)=TEM*(DVEC(1)-XMAT(1)*AV(6))
      CALL WAKMP(IROWA,NR,JY,AV,2)
      RETURN
C
C  MATRIX DOWNSWEEP CALCULATIONS (WAKPY)
C
200   CALL WAKMP(IROWA,NR,JY,XV,1)
      XV(6)=XV(6)-XV(1)*AV(6)
      AV(6)=XV(6)
      CALL WAKMP(IROWT,NR,JY,TV,1)
      TEM=ABS(AV(6))
      IF (TEM-FMAXV(7)) 230,230,220
220   FMAXV(7)=TEM
      YMAXV(7)=YOLDV(JY)
      ZMAXV(7)=ZOLDV(NR)
230   TEM=ABS(AV(6)-TV(6))
      Y=YOLDV(JY)
      Z=ZOLDV(NR)
      DY=YOLDV(JY+1)-Y
      IF (JY-1) 233,233,232
232   DY=DY+Y-YOLDV(JY-1)
233   DZ=ZOLDV(NR+1)-Z
      IF (NR-1) 235,235,234
234   DZ=DZ+Z-ZOLDV(NR-1)
235   DUI=DUI+TEM*TEM*DY*DZ
      DLI=DLI+TV(5)*TV(5)*DY*DZ
      IF (TEM-TMAXV(7)) 250,250,240
240   TMAXV(7)=TEM
250   TV(6)=AV(6)
      CALL WAKMP(IROWT,NR,JY,TV,2)
      IF (LYFAF) 260,260,252
252   IF (JY-LYFAF) 260,254,256
254   VALUE(NR,1)=TV(6)
      RETURN
256   IF (JY-LYFAF+1) 260,258,260
258   VALUE(NR,2)=TV(6)
260   RETURN
      END
CART ID 0105  DB ADDR 4240   DB CNT 00C2
```

```
WAKPP.S(0105)
**WAKPP - STRATIFIED SUBMARINE WAKE, PRINTER PLOTTER ROUTINE
      SUBROUTINE WAKPP
C
C  THIS SUBROUTINE IN THE WAKE PROGRAM SCANS THE WORKING FILE
C  AND CONSTRUCTS A POINT PLOT FOR THE PRINTER
C
      DIMENSION II(21),JVEC(101)
      DIMENSION XVEC(40),JVCHV(7),JSCAL(101)
*COPY (CMWAK)
      EQUIVALENCE (JVEC(2),YNEWV(1)),(XVEC(1),ZNEWV(1))
C
      DATA IH/1H-/,IV/1HI/,IO/1HO/
      DATA II/1HM,1HM,1H ,1H8,1H ,1H6,1H ,1H4,1H ,1H2,1H ,1H1,1H ,
     1 1H3,1H ,1H5,1H ,1H7,1H ,1HP,1HP/
      DATA JVCHV/2HQQ,2HRO,2HU ,2HV ,2HW ,2HSL,2HP /
C
1005  FORMAT(1H1)
1010  FORMAT(18H PRINTER PLOT FOR ,A2,5X,11HX POINTS = ,I3,5X,4HX = ,
     1 E11.5,5X,16HABS MAX VALUE = ,E11.5,5X,13HEDGE VALUE = ,E12.5)
1020  FORMAT(20X,101A1)
1040  FORMAT(12HOPERCENT MAX,3X,6H90/100,5X,5H70/80,5X,5H50/60,5X,
     1 5H30/40,5X,5H10/20,5X,4H1/-1,5X,7H-10/-20,3X,7H-30/-40,3X,
     2 7H-50/-60,3X,7H-70/-80,2X,8H-90/-100/12H NOTATION    ,5X,1HP,
     3 10X,1H7,9X,1H5,9X,1H3,9X,1H1,8X,1H0,10X,1H2,9X,1H4,9X,1H6,9X,
     4 1H8,8X,1HM/)
1050  FORMAT(E18.5,1X,103A1)
1060  FORMAT(19X,103A1)
1070  FORMAT(6X,6E20.5)
1080  FORMAT(5X,1HZ,13X,103A1)
1090  FORMAT(/70X,1HY)
C
C  INITIALIZE PLOT VALUES
C
      IOLAY=9
      IF (LPRFL) 10,350,350
10    IF (LTOLF) 15,20,15
15    IF (LOUT) 20,50,30
20    CALL DATSW(7,ISW)
      GO TO (50,350),ISW
30    CALL DATSW(3,ISW)
      GO TO (50,40),ISW
40    LTOLF=0
50    MOOD=1
      NY=101
      NZ=51
      NYPSX=NYPS-1
      NYPEX=NYPE+1
      NRSTX=NRST-1
      NREND=NRST+JEND
C
C  SET UP AXES AND NOTATION
C
      YMIN=YOLDV(NYPSX)
      YMAX=YOLDV(NYPEX)
      ZMIN=ZOLDV(NRSTX)
      ZMAX=ZOLDV(NREND)
      DY=(YMAX-YMIN)/FLOAT(NY-1)
      DZ=(ZMAX-ZMIN)/FLOAT(NZ-1)
```

```
      DO 80 I=1,6
      YMAT(I)=YMIN+20.0*FLOAT(I-1)*DY
      ZMAT(I)=ZMIN+10.0*FLOAT(I-1)*DZ
80    CONTINUE
      K=19
      DO 90 I=1,NY
      K=K+1
      IF (K-20) 84,82,82
82    K=0
      JSCAL(I)=IV
      GO TO 90
84    JSCAL(I)=IH
90    CONTINUE
C
C  LOOP THROUGH FILE FOR EACH VARIABLE
C
      IROWR=2
      IF (LBURF) 92,94,92
92    IROWR=4
94    WRITE (NOUT,1005)
      DO 300 IVAR=1,NVART
      FMAX=FMAXV(IVAR)
      FMIN=1.0E-04*FMAX
      IF (FMAX) 300,300,100
100   JVAR=IVAR
      IF (IVAR-NVART) 102,101,101
101   JVAR=6
      IROWR=IROWT
102   WRITE (NOUT,1010) JVCHV(IVAR),NPTSN,XP,FMAX,ZEROV(IVAR)
      WRITE (NOUT,1040)
      WRITE (NOUT,1020) JSCAL
      KS=9
      IS=7
      NS=NZ/2+1
      NR=NREND-1
      NPOS=NR
      CALL WAKRR(NR+1,ZPV)
      NPOS=NR+1
      CALL WAKRR(NR,ZMV)
C
C  LOCATE EVERY DESIRED Z VALUE
C
      DO 250 IZ=1,NZ
      JZ=NZ-IZ+1
      Z=ZMIN+FLOAT(JZ-1)*DZ
105   IF (Z-ZP) 107,110,106
106   Z=ZP
      GO TO 110
107   IF (Z-ZM) 108,110,110
108   IF (NR-NRSTX) 1095,1095,109
109   NR=NR-1
      CALL WAKMR(1,3)
      CALL SFVMV(ZM,ZP,NWWZP
      NPOS=NR+1
      CALL WAKRR(NR,ZMV)
      GO TO 105
1095  Z=ZM
C
C  INTERPOLATE FOR PARAMETER VALUES
```

```
C
110   RATZ=(Z-ZM)/(ZP-ZM)
      DO 115 JY=NYPSX,NYPEX
      CALL WAKMP(IROWR,NR,JY,XMYV,1)
      CALL WAKMP(IROWR,NR+2,JY,XPYV,1)
      XVEC(JY)=XMYV(JVAR)+RATZ*(XPYV(JVAR)-XMYV(JVAR))
115   CONTINUE
C
C  EXPAND TO FILL PLOT ARRAY
C
      K=NYPSX
      DO 160 JY=1,NY
      Y=YMIN+FLOAT(JY-1)*DY
116   IF (Y-YOLDV(K)) 119,120,117
117   IF (Y-YOLDV(K+1)) 120,120,118
118   K=K+1
      IF (K-NYPEX) 116,119,119
119   TEM=XVEC(K)
      GO TO 121
120   RATY=(Y-YOLDV(K))/(YOLDV(K+1)-YOLDV(K))
      TEM=XVEC(K)+RATY*(XVEC(K+1)-XVEC(K))
C
C  CHECK AND INSERT POINT VALUES
C
121   I=IFIX(10.0*TEM/FMAX)
      JVEC(JY)=II(I+11)
      IF (I) 160,122,160
122   I=IFIX(100.0*TEM/FMAX)
      IF (I) 160,124,160
124   IF (ABS(TEM)-FMIN) 160,126,126
126   JVEC(JY)=IO
160   CONTINUE
C
C  PRINT TO PRINTER
C
      KS=KS+1
      IF (KS-10) 204,202,202
202   KS=0
      IS=IS-1
      WRITE (NOUT,1050) ZMAT(IS),IH,JVEC,IH
      GO TO 250
204   IF (JZ-NS) 208,206,208
206   WRITE (NOUT,1080) IV,JVEC,IV
      GO TO 250
208   WRITE (NOUT,1060) IV,JVEC,IV
250   CONTINUE
      WRITE (NOUT,1020) JSCAL
      WRITE (NOUT,1070) (YMAT(I),I=1,6)
      WRITE (NOUT,1090)
300   CONTINUE
350   RETURN
      END
CART ID 0105  DB ADDR 48C0   DB CNT 0152
```

```
WAKPY.S(0105)
**WAKPY - STRATIFIED SUBMARINE WAKE, PRESSURE STEP IN Y
      SUBROUTINE WAKPY
C
C  THIS SUBROUTINE IN THE WAKE PROGRAM MAKES A STEP OF DELTA XX/2
C  WITH Z DERIVATIVES EVALUATED AT THE PRESENT XX POINT
C
*COPY (CMWAK)
C
C  INITIALIZE COMPUTATION
C
      IYPSN=IYPS
      IYPEN=IYPE
C
C  INITIALIZE FOR THE Y DIRECTION AND SWEEP
C
      Y=YOLDV(IYPSN-1)
      YP=YOLDV(IYPSN)
      CALL WAKMP(IROWR,NR+1,IYPSN-1,XPYV,1)
      CALL WAKMP(IROWR,NR,IYPSN-1,XV,1)
      CALL WAKMP(IROWT,NR,IYPSN-1,TV,1)
      IF (JZ) 1081,1081,1082
1081  CALL WAKRF(XPYV,XMYV,4)
      GO TO 1083
1082  CALL WAKMP(IROWR,NR-1,IYPSN-1,XMYV,1)
1083  CALL WAKMP(IROWR,NR,IYPSN,XZPV,1)
      JY=IYPSN-1
      IF (LYLFF) 109,110,110
109   CALL WAKRF(XZPV,XZMV,3)
      YM=Y+Y-YP
      GO TO 120
C
C  UPWARD PASS
C
110   JY=JY+1
      CALL SFVMV(XV,XZMV,NMOVE)
      CALL WAKMP(IROWT,NR,JY,TV,1)
      YM=Y
      Y=YP
      YP=YOLDV(JY+1)
      CALL WAKMP(IROWR,NR+1,JY,XPYV,1)
      IF (JZ) 116,116,118
116   CALL WAKRF(XPYV,XMYV,4)
      GO TO 119
118   CALL WAKMP(IROWR,NR-1,JY,XMYV,1)
119   CALL WAKMP(IROWR,NR,JY+1,XZPV,1)
C
C  CALCULATE MATRIX COEFFICIENTS AND INVERT FOR GAMMA AND AV
C
120   CALL WAKPM(1)
      IF (JY-IYPEN) 110,125,125
C
C  UPPER BOUNDARY CONDITION
C
125   TV(6)=AV(6)
      CALL WAKMP(IROWT,NR,JY,TV,2)
C
C  INITIALIZE FOR DOWNWARD PASS
C
```

```
      IYPEN=JY
      IYPEX=IYPEN-1
      IYPSX=IYPSN+LYLFF
      IF (IYPSX-IYPEN) 150,156,156
150   DO 151 IY=IYPSX,IYPEX
      JY=IYPEX+IYPSX-IY
      CALL WAKPM(2)
151   CONTINUE
C
C  ROW SWEEP COMPLETED
C
156   IYPS=IYPSN
      IYPE=IYPEN
      CALL WAKWR(NR,ZV)
      RETURN
      END
CART ID 0105  DB ADDR 3230   DB CNT 0094
```

```
WAKPZ.S(0105)
**WAKPZ - STRATIFIED SUBMARINE WAKE, PRESSURE STEP IN Z
      SUBROUTINE WAKPZ
C
C  THIS SUBROUTINE IN THE WAKE PROGRAM MAKES A STEP OF DELTA XX/2
C  WITH Y DERIVATIVES EVALUATED AT THE PRESENT XX POINT
C
*COPY (CMWAK)
C
C  INITIALIZE COMPUTATION
C
      LZFAF=2
      IROWR=1
      IROWA=3
      NRSTN=NRST
      JENDN=JEND
      NR=NRSTN-1
      NPOS=NR
      CALL WAKRR(NR,ZV)
      CALL WAKRR(NR+1,ZPV)
      JZ=0
      IF (LZLFF) 101,1015,1015
101   ZM=Z+Z-ZP
      IYPEM=IYPEP
      IYPSM=IYPSP
      GO TO 110
1015  NR=NRSTN
102   JZ=JZ+1
      NPOS=NR
      CALL SFVMV(Z,ZM,NWWZF)
      CALL WAKMR(2,1)
      CALL SFVMV(ZP,Z,NWWZF)
      CALL WAKMR(3,2)
      CALL WAKRR(NR+1,ZPV)
C
C  INITIALIZE FOR THE Y DIRECTION AND SWEEP
C
110   Y=YOLDV(IYPS-1)
      YP=YOLDV(IYPS)
      CALL WAKMP(IROWR,NR+1,IYPS-1,XPYV,1)
      CALL WAKMP(IROWR,NR,IYPS-1,XV,1)
      CALL WAKMP(IROWT,NR,IYPS-1,TV,1)
      IF (JZ) 111,111,1115
111   CALL WAKRF(XPYV,XMYV,4)
      GO TO 112
1115  CALL WAKMP(IROWR,NR-1,IYPS-1,XMYV,1)
112   IYPSX=IYPS+LYLFF
      IF (JZ) 1126,1126,1125
1125  CALL WAKMP(IROWA,NR-1,IYPS-1,AV,1)
1126  CALL WAKMP(IROWR,NR,IYPS,XZPV,1)
      DO 145 JY=IYPSX,IYPE
      IF (JY-IYPS) 114,1142,1142
114   CALL WAKRF(XZPV,XZMV,3)
      YM=Y+Y-YP
      GO TO 120
1142  IF (JZ) 1145,1145,1144
1144  CALL WAKMP(IROWA,NR-1,JY,AV,1)
1145  CALL SFVMV(XV,XZMV,NMOVE)
      CALL WAKMP(IROWT,NR,JY,TV,1)
```

```
      YM=Y
      Y=YP
      YP=YOLDV(JY+1)
      CALL WAKMP(IROWR,NR+1,JY,XPYV,1)
      IF (JZ) 1148,1148,1149
1148  CALL WAKRF(XPYV,XMYV,4)
      GO TO 119
1149  CALL WAKMP(IROWR,NR-1,JY,XMYV,1)
119   CALL WAKMP(IROWR,NR,JY+1,XZPV,1)
C
C  CALCULATE MATRIX COEFFICIENTS AND INVERT FOR GAMMA AND AV
C
120   CALL WAKPM(1)
145   CONTINUE
      IF (JZ-JENDN+1) 146,147,147
146   CALL WAKWR(NR,ZV)
147   NR=NR+1
      IF (JZ-JENDN) 102,150,150
C
C  INITIALIZE FOR DOWNWARD PASS
C
150   NR=NR-1
      JENDN=JZ
      IZEND=JENDN-LZLFF-1
      LZFAF=1
      IROWR=3
      IROWA=4
      DO 154 IZ=1,IZEND
      IYEND=IYPE-IYPS-LYLFF+1
      DO 1508 IY=1,IYEND
      JY=IYPE-IY+1
      IF (JY-IYPEM) 1502,1502,1508
1502  IF (JY-IYPSM) 1503,1506,1506
1503  IF (IYPSM-2) 1506,1506,1508
1506  CALL WAKMP(IROWR,NR,JY,AV,1)
      CALL WAKMP(IROWR,NR-1,JY,XV,1)
      XV(6)=XV(6)-XV(1)*AV(6)
      CALL WAKMP(IROWR,NR-1,JY,XV,2)
1508  CONTINUE
C
C  ROW SWEEP COMPLETED
C
      CALL WAKPY
      NR=NR-1
      NPOS=NR
      JZ=JZ-1
      CALL SFVMV(Z,ZP,NWWZF)
      CALL WAKMR(2,3)
      CALL SFVMV(ZM,Z,NWWZF)
      CALL WAKMR(1,2)
      IF (IZ-IZEND) 153,152,152
152   IF (LZLFF) 154,153,153
153   CALL WAKRR(NR-1,ZMV)
154   CONTINUE
      IF (JZ) 156,156,158
156   ZM=Z+Z-ZP
      IYPEM=IYPEP
      IYPSM=IYPSP
158   CALL WAKPY
```

```
      JEND=JENDN
      NRST=NRSTN
      RETURN
      END
CART ID 0105  DB ADDR 3130   DB CNT 00F2
```

```
WAKRF.S(0105)
**WAKRF - STRATIFIED SUBMARINE WAKE, REFLECT A POINT
      SUBROUTINE WAKRF(TEMA,TEMB,LFL)
C
C  THIS SUBROUTINE IN THE WAKE PROGRAM PROVIDES REFLECTION
C
      DIMENSION TEMA(6),TEMB(6)
*COPY (CMWAK)
C
      CALL SFVMV(TEMA,TEMB,NWVEC)
      DO 100 I=1,NWVEC
      IF (IRFV(I,LFL)) 100,50,100
50    TEMB(I)=-TEMB(I)
      GO TO (60,60,100,100),LFL
60    TEMB(I)=TEMB(I)+2.0*ZEROV(I)
100   CONTINUE
      RETURN
      END
CART ID 0105  DB ADDR 4A20   DB CNT 0022
```

```
WAKRR.S(0105)
**WAKRR - STRATIFIED SUBMARINE WAKE, READ A Z BUFFER ROW
      SUBROUTINE WAKRR(NRX,ZPOS)
C
C  THIS SUBROUTINE IN THE WAKE PROGRAM READS A Z ROW AND
C  FILLS THE ROW BUFFER AT THE NRX POSITION FOR ALL Y
C
      DIMENSION ZPOS(2),RECB(24)
*COPY (CMWAK)
      EQUIVALENCE (RECB(1),DVEC(1))
C
      DATA JERRX/2HRR/
C
C  FILL ZPOS WITH IYPS AND IYPE VALUES
C
      IF (NRX) 300,300,10
10    IF (NRX-MXRZ) 20,20,300
20    ZA=ZOLDV(NRX)
      IVECA=IYPSV(NRX)
      IVECB=IYPEV(NRX)
      CALL SFVMV(ZA,ZPOS,NWWZF)
C
C  LOCATE NRX POSITION IN ROW BUFFER ARRAY
C
      J=NRX-NPOS+2
      IF (J) 300,300,30
30    IF (J-3) 40,40,300
40    IF (IVECA) 45,45,50
45    IYPSX=1
      GO TO 140
50    IYPSX=IVECA+LYLFF
      IF (IYPSX-1) 90,90,60
60    IYPEX=IYPSX-1
C
C  FILL BUFFER ROW WITH LEADING ZEROES
C
      DO 80 I=1,IYPEX
      CALL SFVFL(0.0,ROWB(1,I,J),NWVEC)
      CALL SFVMV(ZEROV,ROWB(7,I,J),NWVEC)
      CALL SFVMV(ZEROV,ROWB(13,I,J),NWVEC)
80    CONTINUE
90    IYPEX=IVECB
C
C  FILL BUFFER ROW WITH NRX INFORMATION
C
      NRXX=(NRX-1)*MXRY+IYPSX
      DO 130 I=IYPSX,IYPEX
      CALL PBFDR(SLNID,NRXX,NWR,RECB)
      CALL SFVMV(RECB(1),ROWB(1,I,J),NWVEC)
      IF (MOOD) 105,110,115
105   CALL SFVMV(RECB(7),ROWB(7,I,J),NMOVE)
      GO TO 130
110   CALL SFVMV(RECB(13),ROWB(7,I,J),NMOVE)
      GO TO 130
115   CALL SFVMV(RECB(7),ROWB(7,I,J),NWVEC)
      CALL SFVMV(RECB(19),ROWB(13,I,J),NWVEC)
130   CONTINUE
      IYPSX=IYPEX+1
      IF (IYPSX-MXRY) 140,140,160
```

```
C
C  FILL BUFFER ROW WITH TRAILING ZEROES
C
140    DO 150 I=IYPSX,MXRY
       CALL SFVFL(0.0,ROWB(1,I,J),NWVEC)
       CALL SFVMV(ZEROV,ROWB(7,I,J),NWVEC)
       CALL SFVMV(ZEROV,ROWB(13,I,J),NWVEC)
150    CONTINUE
160    RETURN
300    JERR=JERRX
       RETURN
       END
CART ID 0105   DB ADDR 4A50   DB CNT 008C
```

```
WAKSC.S(0105)
**WAKSC - STRATIFIED SUBMARINE WAKE, SUPEREQUILIBRIUM CALCULATIONS
      SUBROUTINE WAKSC
C
C  THIS SUBROUTINE IN THE WAKE PROGRAM COMPUTES THE AUXILIARY
C  TURBULENCE VIA SUPEREQUILIBRIUM THEORY
C
C   1) FOR THE RIGHT HAND SIDE OF POISSONS EQUATION
C
C   2) FOR DU/DY, DU/DZ, DRHO/DY AND DRHO/DZ
C
C   3) FOR TURBULENCE OUTPUT TO THE LINE PRINTER
C
      DIMENSION TURB(10),TOTAL(560)
*COPY (CMWAK)
      EQUIVALENCE (TURB(1),YMAT(1)),(DIVT,GM(11))
      EQUIVALENCE (FF,GM(13)),(FT,GM(14))
C
1000  FORMAT(//36H SUPEREQUILIBRIUM ERROR HAS OCCURRED,2I5,3E15.5)
1001  FORMAT(//42H SUPEREQUILIBRIUM CORRECTION TO ZERO NOISE,I4,
     1 14H TIMES AT STEP,I4)
1002  FORMAT(//16X,7HTURB KE,8X,7HMEAN KE,8X,7HRHO*RHO,6X,
     1 11HRHO*RHO OUT,4X,11HMEAN KE OUT,8X,4HAREA,
     2 9X,7HPSI MAX,6X,11HDISSIPATION/10X,8E15.5)
1003  FORMAT(1H0,15X,7HF TOTAL,9X,5HF MAX,8X,8HF CHANGE,7X,
     1 10HDIVG ERROR,7X,4HLIFT/10X,5E15.5)
1004  FORMAT(//37H TURBULENCE CORRECTION FOR MAX BOUNDS,2I5,
     1 8H AT STEP,I4)
C
      FPRZ(ARGM,ARG,ARGP)=FZM*(ARGM+FZS*ARG-FZR*ARGP)
      FPRY(ARGM,ARG,ARGP)=FYM*(ARGM+FYS*ARG-FYR*ARGP)
      SPRZ(ARGM,ARG,ARGP)=SZM*ARGM+SZ*ARG+SZP*ARGP
      SPRY(ARGM,ARG,ARGP)=SYM*ARGM+SY*ARG+SYP*ARGP
C
C  ZERO PERTINENT VARIABLES
C
      IOLAY=7
      IF (LSTFL) 300,50,50
50    MOOD=1
      LFL=0
      LVV=0
      LWW=0
      LZFAF=0
      PLANE=1.0/FLOAT(LYLFF+2)/FLOAT(LZLFF+2)
      DEPSN=0.0
      AREA=0.0
      DIVT=0.0
      CVS=0.0
      XFI=0.0
      FF=0.0
      FT=0.0
      CALL SFVFL(0.0,TURBX,10)
      CALL SFVMV(ROWG,TOTAL,560)
      SCMX=4.0*YSCAL*YSCAL*ZSCAL/(YSCAL*YSCAL+ZSCAL*ZSCAL)
      ZEROV(6)=SCMX
      FMAXV(6)=SCMX
C
C  INITIALIZE FOR PASSING THROUGH PROFILE
C
```

```
      IROWA=2
      IROWR=4
      IF (LSTFL) 96,96,95
95    IROWR=2
96    NRSTX=NRST+LZLFF
      NREND=NRST+JEND-1
      NPOS=NRST-1
      CALL WAKRR(NRST-1,ZV)
      CALL WAKRR(NRST,ZPV)
      DO 250 NR=NRSTX,NREND
C
C  READ THREE SURROUNDING ROWS AND TEST FOR SELECTED Z IN DOMAIN
C
      IF (NR-NRST) 104,103,103
103   NPOS=NR
      CALL SFVMV(Z,ZM,NWWZF)
      CALL WAKMR(2,1)
      CALL SFVMV(ZP,Z,NWWZF)
      CALL WAKMR(3,2)
      CALL WAKRR(NR+1,ZPV)
      GO TO 105
104   ZM=Z+Z-ZP
105   DZM=Z-ZM
      DZP=ZP-Z
      DZT=ZP-ZM
      CALL WAKDG(Z)
C
C  STEP THROUGH ALL Y POINTS COMPUTING AUXILIARY QUANTITIES
C
      Y=YOLDV(IYPS-1)
      YP=YOLDV(IYPS)
      CALL WAKMP(IROWR,NR+1,IYPS-1,XPYV,1)
      CALL WAKMP(IROWR,NR,IYPS-1,XV,1)
      IF (NR-NRST) 108,109,109
108   CALL WAKRF(XPYV,XMYV,2)
      GO TO 110
109   CALL WAKMP(IROWR,NR-1,IYPS-1,XMYV,1)
110   IYPSX=IYPS+LYLFF
      CALL WAKMP(IROWT,NR,IYPS-1,TV,1)
      CALL WAKMP(IROWR,NR+1,IYPS,XPPV,1)
      CALL WAKMP(IROWR,NR,IYPS,XZPV,1)
      IF (NR-NRST) 113,114,114
113   CALL WAKRF(XPPV,XMPV,2)
      GO TO 115
114   CALL WAKMP(IROWR,NR-1,IYPS,XMPV,1)
115   DO 200 JY=IYPSX,IYPE
      IF (JY-IYPS) 116,117,117
116   CALL WAKRF(XPPV,XPMV,1)
      CALL WAKRF(XZPV,XZMV,1)
      CALL WAKRF(XMPV,XMMV,1)
      YM=Y+Y-YP
      GO TO 130
117   CALL WAKMP(IROWT,NR,JY,TV,1)
      CALL SFVMV(XPYV,XPMV,NMOVE)
      CALL SFVMV(XV,XZMV,NMOVE)
      CALL SFVMV(XMYV,XMMV,NMOVE)
      YM=Y
      Y=YP
      YP=YOLDV(JY+1)
```

```
      CALL WAKMP(IROWR,NR+1,JY+1,XPPV,1)
      CALL WAKMP(IROWR,NR,JY+1,XZPV,1)
      IF (NR-NRST) 125,128,128
125   CALL WAKRF(XPPV,XMPV,2)
      GO TO 130
128   CALL WAKMP(IROWR,NR-1,JY+1,XMPV,1)
130   DYM=Y-YM
      DYP=YP-Y
      DYT=YP-YM
C
C  COMPUTE SPACING FACTORS
C
      FZM=-DZP/DZM/DZT
      FZP=DZM/DZP/DZT
      FZ=-FZM-FZP
      TEM=DZM/DZP
      FZR=TEM*TEM
      FZS=FZR-1.0
      SZM=2.0/DZM/DZT
      SZP=2.0/DZP/DZT
      SZ=-SZM-SZP
C
      FYM=-DYP/DYM/DYT
      FYP=DYM/DYP/DYT
      FY=-FYM-FYP
      TEM=DYM/DYP
      FYR=TEM*TEM
      FYS=FYR-1.0
      SYM=2.0/DYM/DYT
      SYP=2.0/DYP/DYT
      SY=-SYM-SYP
C
C  CHECK APPROPRIATE DENOMINATOR VALUES
C
      IF (XV(1)) 135,141,141
135   XV(1)=0.0
      LFL=LFL+1
141   Q=SQRT(XV(1))
      FRZ=FPRZ(XMYV(2),XV(2),XPYV(2))
      FTEM=FRZ+DRDZ
      CALL WAKCL(XV,FTEM)
      IF (XV(6)-SCALM) 1410,1411,1411
1410  XV(6)=SCALM
1411  IF (XV(6)-SCMX) 1420,1420,1412
1412  XV(6)=SCMX
1420  IF (LAMIN) 142,143,142
142   Q=0.0
      GO TO 150
143   BQL=BBETA*Q/SCALE
      C1=A*BQL*Q/SCALE-G*FTEM
      C2=BBS*G*FTEM-C1
      IF (C1*C2) 150,144,150
144   IF (Q) 145,150,145
145   WRITE (NOUT,1000) NR,JY,XV(1),C1,C2
      LTRNF=1
      RETURN
C
C  COMPUTE DERIVATIVES AND RIGHT HAND SIDE
C
```

```
150   FRY=FPRY(XZMV(2),XV(2),XZPV(2))
      FUZ=FPRZ(XMYV(3),XV(3),XPYV(3))
      FUY=FPRY(XZMV(3),XV(3),XZPV(3))
      TV(1)=FUY
      TV(2)=FUZ
      TV(3)=FRY
      TV(4)=FRZ
      IF (Q) 153,153,152
152   TEM=FUY*FUY*SCALE/Q/BQL-BBETA*G*FTEM/C2
     1 -FUZ*FUZ*BBETA*(G*FTEM*(1.0-A/B/S)+A*A*XV(1)/SCALE/SCALE)/C1/C2
      CTEM=CVV+(B-CVV*TEM)/(HBETA+TEM)
      CALL WAKTC(CTEM,1.0,LVV)
153   IF (NSTAT-1) 1540,1535,1540
1535  TV(5)=0.0
      TV(6)=0.0
      FVZ=0.0
      FWY=0.0
      GO TO 160
1540  TEMF=TV(5)
      CALL WAKLL(TEMK,2)
      IF (Q) 154,154,155
154   CTEM=0.0
      TEM=0.0
      TEMC=0.0
      GO TO 156
155   C3=CTEM*(DRDZ+G*BBS*FRY*FRY/C1)
      TEM=2.0*CTEM*G*FRY*FPRZ(FPRY(XMMV(1),XMYV(1),XMPV(1)),
     1 FPRY(XZMV(1),XV(1),XZPV(1)),FPRY(XPMV(1),XPYV(1),XPPV(1)))/C1
      TEMC=CTEM-2.0*G*(C3+CTEM*FRZ)/C2
      CALL WAKTC(TEMC,1.0-CTEM,LWW)
      TEMC=TEMC*SPRZ(XMYV(1),XV(1),XPYV(1))
156   FVY=FPRY(XZMV(4),XV(4),XZPV(4))
      FVZ=FPRZ(XMYV(4),XV(4),XPYV(4))
      FWY=FPRY(XZMV(5),XV(5),XZPV(5))
      FWZ=FPRZ(XMYV(5),XV(5),XPYV(5))
      TEMA=2.0*FVZ*FWY
      TEMB=2.0*FVY*FWZ
      TV(5)=-TEMC-CTEM*SPRY(XZMV(1),XV(1),XZPV(1))-TEM-TEMA+TEMB-G*FRZ
      DXX=DX
      IF (DXX-DIVP*DXMAX) 157,158,158
157   DXX=DIVP*DXMAX
158   DXX=DIVF*DXX
      TV(5)=TV(5)+U*(FVY+FWZ)/DXX-TEMK
160   CALL WAKMP(IROWT,NR,JY,TV,2)
C
C  WRITE COMPLETED SOLUTION FOR NEXT INTEGRATION STEP
C
      CALL WAKMP(IROWA,NR,JY,XV,2)
C
C  COMPUTE RICHARDSON SCALE LENGTH COMPARISON
C
      IF (YOUT-YM) 170,161,1605
1605  IF (YOUT-Y) 161,170,170
161   IF (LZFAF) 170,162,170
162   IF (4.0*XV(1)-FMAXV(1)) 163,163,170
163   CWS=ABS(FTEM)
      LZ .F=1
C
C  COMPUTE SUPEREQUILIBRIUM TURBULENCE VALUES
```

```
C
170   IF (Q) 171,171,172
171   CALL SFVFL(0.0,TURB,10)
      GO TO 176
C
C  ORDER - UV  UW  VW  UR  VR  WR  RR  UU  VV  WW
C
172   TURB(9)=CTEM*XV(1)
      TURB(3)=TURB(9)*G*FRY/C1
      TURB(5)=-BQL*TURB(3)/G
      TURB(1)=-(TURB(9)*FUY+TURB(3)*FUZ)/BQL
      TURB(6)=(BQL*(TURB(9)*FTEM+TURB(3)*FRY)
     1 -TURB(5)*BBETA*G*FRY/B/S)/C2
      TURB(10)=TURB(9)-2.0*G*TURB(6)/BQL
      CALL WAKTC(TURB(10),XV(1)-TURB(9),I)
      TURB(8)=XV(1)-TURB(9)-TURB(10)
      TURB(7)=-SCALE*(TURB(5)*FRY+TURB(6)*FTEM)/Q/B/S
      TURB(4)=(FTEM*(TURB(3)*FUY+TURB(10)*FUZ)
     1 -BQL*(TURB(1)*FRY+TURB(5)*FUY+TURB(6)*FUZ))/C1
      TURB(2)=-(G*TURB(4)+TURB(3)*FUY+TURB(10)*FUZ)/BQL
      TURB(3)=TURB(3)-Q*SCALE*(FWY+FVZ)/3.0
      DO 174 I=1,10
      IF (ABS(TURB(I))-ABS(TURBX(I))) 174,174,173
173   TURBX(I)=TURB(I)
174   CONTINUE
      IF (CTEM-CVS) 176,176,175
175   CVS=CTEM
176   IF (LSOLF) 178,180,178
178   CALL WAKMP(IROWG,NR,JY,TURB,2)
C
C  COMPUTATION OF INTEGRALS
C
180   SUMF=PLANE*DYT*DZI
      IF (NR-NRST) 181,182,182
181   SUMF=0.5*SUMF
182   IF (JY-NYPS) 183,184,184
183   SUMF=0.5*SUMF
184   DEPSN=DEPSN+Q*XV(1)*SUMF/SCALE
      IF (NSTAT-1) 185,200,185
185   XFI=XFI+TV(5)*SUMF
      AREA=AREA+SUMF
      TEM=FVY+FWZ
      DIVT=DIVT+TEM*TEM
      IF (ABS(TV(5))-ABS(FF)) 187,187,186
186   FF=TV(5)
187   TEM=ABS(TV(5)-TEMF)
      IF (TEM-FT) 200,200,188
188   FT=TEM
200   CONTINUE
      CALL WAKWR(NR,ZV)
      IF (LSOLF) 205,250,205
205   CALL WAKMG(NR,ZV,2)
250   CONTINUE
      CALL SFVMV(TOTAL,ROWG,560)
C
C  COMPUTE AND PRINT PERTINENT INTEGRALS
C
      EPSSV(2)=SQRT(RIS*FMAXV(1)/G/CWS)/2.0
      IF (LVV+LWW) 252,252,251
```

```
251    WRITE (NOUT,1004) LVV,LWW,NPTSN
252    IF (LFL) 260,260,255
255    WRITE (NOUT,1001) LFL,NPTSN
260    DEPST=DEPST+0.5*DXSAV*(DEPSI+DEPSN)*B
       DEPSI=DEPSN
       WRITE (NOUT,1002) (GM(I),I=1,7),DEPST
       IF (NSTAT-1) 270,300,270
270    IF (FMAXV(4)*FMAXV(5)) 271,272,271
271    DIVT=DIVT*YSCAL*ZSCAL/(AREA*FMAXV(4)*FMAXV(5))
272    WRITE (NOUT,1003) XFI,FF,FT,DIVT,GM(8)
300    RETURN
       END
CART ID 0105  DB ADDR 4AE0   DB CNT 0264
```

```
WAKSE.S(0105)
**WAKSE - STRATIFIED SUBMARINE WAKE, SET EPSILON CRITERION
      SUBROUTINE WAKSE(EPS,EPSV)
C
C  THIS SUBROUTINE IN THE WAKE PROGRAM TRANSFERS EPS TO EPSV
C
      DIMENSION EPSV(5)
*COPY (CMWAK)
C
      DO 110 I=1,NVAR
      TEM=1.0E10
      VWTF=VWTFV(I)
      IF (VWTF) 108,108,100
100   VMAX=GMAXV(I)
      VSCA=VSCAV(I)
      IF (VMAX-VSCA*EPSN) 101,102,102
101   VMAX=VSCA*EPSN
102   IF (EPS) 103,104,104
103   TEM=-EPS*VSCA
      GO TO 106
104   TEM=EPS*VMAX
106   TEM=TEM/VWTF
108   EPSV(I)=TEM
110   CONTINUE
      RETURN
      END
CART ID 0105  DB ADDR 2730   DB CNT 0032
```

```
WAKSR,S(0105)
**WAKSR - STRATIFIED SUBMARINE WAKE, SPECIAL RESTART PROFILES
      SUBROUTINE WAKSR(FILID)
C
C  THIS SUBROUTINE IN THE WAKE PROGRAM LOCATES A SET OF
C  SPECIAL RESTART PROFILES AND INITIALIZES THE WORKING FILE
C
      DIMENSION FILID(3),RBUF(6,40),IBUF(40),IVEC(3)
*COPY (CMWAK)
      EQUIVALENCE (RBUF(1,1),ROWG(1,1)),(IBUF(2),ROWG(1,16))
C
      DATA IVEC/3,5,6/
C
1000  FORMAT(//25H IMPROPER SPECIAL RESTART,4I5)
C
C  LOCATE PROPER X RECORD
C
      NR=2
      K=IVEC(LZFAF)
210   NP=NR-1
      CALL PBFOR(FILID,NP,2,ZA)
      NY=IVECA
      NZ=IVECB
      NR=NP+NY+NZ+K*NY*NZ
      CALL PBFOR(FILID,NR,1,XP)
      IF (XP) 215,212,212
212   IF (XP-X) 210,210,215
215   X=ZA
      IF (LZLFF) 2151,2152,2152
2151  NRST=2
      GO TO 2153
2152  NRST=(MXRZ-NZ)/2+2
2153  JEND=NZ-2
      IF (LYLFF) 2154,2155,2155
2154  NYPS=2
      GO TO 2156
2155  NYPS=(MXRY-NY)/2+2
2156  NYPE=NY+NYPS-3
      IF (JEND) 218,218,2160
2160  IF (NRST-2) 218,2165,2165
2165  IF (NYPS-NYPE) 2170,218,218
2170  IF (NYPS-2) 218,220,220
218   WRITE (NOUT,1000) NYPS,NYPE,NRST,JEND
      CALL EXIT
220   CALL PBFOR(FILID,NP,NY,YOLDV(NYPS-1))
      CALL PBFOR(FILID,NP,NZ,ZOLDV(NRST-1))
C
C  BUILD WORKING FILE FROM SPECIAL RESTART PROFILES
C
      CALL SFVFL(0.0,TV,NWVEC)
      DO 250 JZ=1,NZ
      DO 2200 JY=1,40
      IBUF(JY)=0
      CALL SFVMV(ZEROV,RBUF(1,JY),NVAR)
      RBUF(6,JY)=ZEROV(7)
2200  CONTINUE
      DO 230 IVAR=1,NWVEC
      JVAR=IVAR
      KVAR=IVAR
```

```
      GO TO (221,221,2202,2204,2204,2203),IVAR
 2202 IF (NSTAT-2) 2207,230,2207
 2203 KVAR=KVAR+1
 2204 IF (NSTAT-1) 2206,230,2206
 2206 IF (LZFAF-2) 230,2208,2208
 2207 IF (LZFAF-2) 221,230,221
 2208 JVAR=JVAR+LZFAF-3
 221  NR=NP+(JVAR-1)*NY*NZ
      DO 225 JY=1,NY,10
      I=JY/10+1
      J=10*(I-1)
      NRY=MMIN(10,NY-J)
      NRXX=NR+NRY*(JZ-1)+J*NZ
      CALL PBFDR(FILID,NRXX,NRY,TURBX)
      DO 224 I=1,NRY
      J=J+1
      RBUF(IVAR,J)=TURBX(I)
      TEM=ABS(RBUF(IVAR,J))
      IF (ABS(RBUF(IVAR,J)-ZEROV(KVAR))-1.0E-04) 2214,2214,2215
 2214 IBUF(J)=IBUF(J)+1
 2215 IF (TEM-GMAXV(KVAR)) 224,224,222
 222  GMAXV(KVAR)=TEM
 224  CONTINUE
 225  CONTINUE
 230  CONTINUE
      NR=NRST+JZ-2
      NPOS=NR
      Z=ZOLDV(NR)
      IYPS=0
      IYPE=0
      DO 240 NRY=1,NY
      JY=NYPS+NRY-2
      IF (IYPS) 231,231,233
 231  IF (IBUF(NRY)-K) 232,240,240
 232  IYPS=MMAX(2,JY)
 233  IF (IBUF(NRY)-K) 234,241,241
 234  IF (NSTAT-1) 235,236,235
 235  TV(6)=RBUF(6,NRY)
      CALL WAKMP(IROWT,NR,JY,TV,2)
 236  RBUF(6,NRY)=SCALE
      CALL WAKMP(IROWR,NR,JY,RBUF(1,NRY),2)
 240  CONTINUE
      IF (IYPS) 242,242,218
 241  IYPE=JY-1
 242  CALL WAKWR(NR,ZV)
 250  CONTINUE
      RETURN
      END
CART ID 0105  DB ADDR 57A0   DB CNT 0004
```

```
WAKSY.S(0105)
**WAKSY - STRATIFIED SUBMARINE WAKE, STEP SOLUTION IN Y
      SUBROUTINE WAKSY
C
C  THIS SUBROUTINE IN THE WAKE PROGRAM MAKES A STEP OF DELTA X/2
C  WITH Z DERIVATIVES EVALUATED AT THE PRESENT X POINT
C
*COPY (CMWAK)
C
C  INITIALIZE COMPUTATION
C
      IOLAY=3
      GO TO (10,20),NSS
10    IROWA=3
      IROWR=2
      MOOD=-1
      LPKRF=0
      GO TO 30
20    IROWA=4
      IROWR=3
      MOOD=0
      IF (LPKRF) 180,30,180
30    NR=NRST-1
      NPOS=NR
      CALL WAKRR(NR,ZV)
      CALL WAKRR(NR+1,ZPV)
      JENDX=JEND
      IF (LZLFF) 100,102,102
100   JENDX=JEND+1
      ZM=Z+Z-ZP
      IYPEM=IYPEP
      IYPSM=IYPSP
102   DO 157 JZ=1,JENDX
      JZN=JZ+LZLFF
      IF (JZN) 104,104,103
103   NR=NR+1
      NPOS=NR
      CALL SFVMV(Z,ZM,NWWZF)
      CALL WAKMR(2,1)
      CALL SFVMV(ZP,Z,NWWZF)
      CALL WAKMR(3,2)
      CALL WAKRR(NR+1,ZPV)
104   IYPSN=IYPS
      IYPEN=IYPE
      IBOT=IYPS+LYLFF
      ITOP=IYPE
C
C  INITIALIZE FOR THE Y DIRECTION AND SWEEP
C
106   Y=YOLDV(IYPSN-1)
      YP=YOLDV(IYPSN)
      CALL WAKMP(IROWR,NR+1,IYPSN-1,XPYV,1)
      CALL WAKMP(IROWT,NR+1,IYPSN-1,TPYV,1)
      CALL WAKMP(IROWR,NR,IYPSN-1,XV,1)
      CALL WAKMP(IROWT,NR,IYPSN-1,TV,1)
      IF (JZN) 1081,1081,1082
1081  CALL WAKRF(XPYV,XMYV,2)
      CALL WAKRF(TPYV,TMYV,4)
      GO TO 1083
```

```
1082  CALL WAKMP(IROWR,NR-1,IYPSN-1,XMYV,1)
      CALL WAKMP(IROWT,NR-1,IYPSN-1,TMYV,1)
1083  CALL WAKMP(IROWR,NR+1,IYPSN,XPPV,1)
      CALL WAKMP(IROWR,NR,IYPSN,XZPV,1)
      CALL WAKMP(IROWT,NR,IYPSN,TZPV,1)
      IF (JZN) 1087,1087,1088
1087  CALL WAKRF(XPPV,XMPV,2)
      GO TO 1089
1088  CALL WAKMP(IROWR,NR-1,IYPSN,XMPV,1)
1089  JY=IYPSN-1
      IF (LYLFF) 109,110,110
109   CALL WAKRF(XPPV,XPMV,1)
      CALL WAKRF(XZPV,XZMV,1)
      CALL WAKRF(XMPV,XMMV,1)
      CALL WAKRF(TZPV,TZMV,3)
      YM=Y+Y-YP
      GO TO 120
C
C  UPWARD PASS
C
110   JY=JY+1
      CALL SFVMV(XPYV,XPMV,NMOVE)
      CALL SFVMV(XV,XZMV,NMOVE)
      CALL SFVMV(XMYV,XMMV,NMOVE)
      CALL SFVMV(TV,TZMV,NMOVE)
      YM=Y
      Y=YP
      YP=YOLDV(JY+1)
      CALL WAKMP(IROWT,NR+1,JY,TPYV,1)
      IF (JZN) 112,112,113
112   CALL WAKRF(TPYV,TMYV,4)
      GO TO 114
113   CALL WAKMP(IROWT,NR-1,JY,TMYV,1)
114   CALL WAKMP(IROWR,NR+1,JY+1,XPPV,1)
      CALL WAKMP(IROWR,NR,JY+1,XZPV,1)
      CALL WAKMP(IROWT,NR,JY+1,TZPV,1)
      IF (JZN) 116,116,118
116   CALL WAKRF(XPPV,XMPV,2)
      GO TO 120
118   CALL WAKMP(IROWR,NR-1,JY+1,XMPV,1)
C
C  CALCULATE MATRIX COEFFICIENTS AND INVERT FOR GAMMA AND AV
C
120   CALL WAKMY
      DO 125 I=1,NWVEC
      TEM=1.0/(YMAT(I)-XMAT(I)*GM(I))
      GM(I)=TEM*ZMAT(I)
      AV(I)=TEM*(DVEC(I)-XMAT(I)*AV(I))
125   CONTINUE
      CALL WAKMP(IROWA,NR,JY,AV,2)
      CALL WAKMP(IROWG,NR,JY,GM,2)
      IF (JY-IYPEN) 110,130,130
C
C  UPPER BOUNDARY CONDITION
C
130   CALL WAKEC(I)
      IF (I) 131,140,131
131   IYPEN=JY
      GO TO 150
```

```
C
C  ADD POINTS TO THE RIGHT SIDE WHERE NEEDED
C
140    IF (JY+2-MXRY) 141,141,170
141    IF (JY-NYPE) 110,148,148
148    YOLDV(JY+2)=YP+DFRMX*(YP-Y)
       GO TO 110
C
C  INITIALIZE FOR DOWNWARD PASS
C
150    IYPEX=IYPEN-1
       IYPSX=IYPSN+LYLFF
       IF (IYPSX-IYPEN) 1500,1515,1515
1500   DO 151 IY=IYPSX,IYPEX
       JY=IYPEX+IYPSX-IY
       CALL WAKMP(IROWA,NR,JY,XV,1)
       CALL WAKMP(IROWG,NR,JY,GM,1)
       DO 1507 I=1,NWVEC
       XV(I)=XV(I)-GM(I)*AV(I)
1507   CONTINUE
       CALL SFVMV(XV,AV,NWVEC)
       CALL WAKMP(IROWA,NR,JY,AV,2)
151    CONTINUE
1515   IF (LYLFF) 156,152,152
C
C  LOWER BOUNDARY CONDITION
C
152    DO 154 I=1,NVAR
       IF (ABS(AV(I)-ZEROV(I))-EPSSV(I)) 154,154,160
154    CONTINUE
C
C  RETURN TO MAINLINE
C
156    IYPS=MMAX(2,IYPSN)
       IYPE=IYPEN
       NYPS=MMIN(NYPS,IYPS)
       NYPE=MMAX(NYPE,IYPE)
       CALL WAKWR(NR,ZV)
157    CONTINUE
C
C  MONOTONICITY CHECK TO ZERO VALUES
C
       NRSTX=NRST+LZLFF
       NREND=NRST+JEND-1
       IF (LYLFF) 1585,1570,1570
1570   I=0
       LL=LZLFF
       IYPS=LZLFF*(MXRY-NYPS)+MXRY
       DO 1580 JZ=NRSTX,NREND
       IF (LL) 1571,1574,1574
1571   IF (IYPSV(JZ)-IYPS) 1572,1580,1579
1572   I=1
       IYPE=IYPS-1
       IYPS=IYPSV(JZ)
       DO 1573 JY=IYPS,IYPE
       CALL WAKWZ(JZ-1,JY)
1573   CONTINUE
       IYPSV(JZ-1)=IYPS
       GO TO 1580
```

```
1574  IF (IYPSV(JZ)-NYPS) 1578,1578,1575
1575  IF (IYPSV(JZ)-IYPS) 1579,1580,1576
1576  I=1
      IYPE=IYPSV(JZ)-1
      DO 1577 JY=IYPS,IYPE
      CALL WAKWZ(JZ,JY)
1577  CONTINUE
      IYPSV(JZ)=IYPS
      GO TO 1580
1578  LL=-1
1579  IYPS=IYPSV(JZ)
1580  CONTINUE
      IF (I) 1585,1585,1570
1585  I=0
      LL=LZLFF
      IYPE=LZLFF*(1-NYPE)+1
      DO 1595 JZ=NRSTX,NREND
      IF (LL) 1586,1589,1589
1586  IF (IYPEV(JZ)-IYPE) 1594,1595,1587
1587  I=1
      IYPS=IYPE+1
      IYPE=IYPEV(JZ)
      DO 1588 JY=IYPS,IYPE
      CALL WAKWZ(JZ-1,JY)
1588  CONTINUE
      IYPEV(JZ-1)=IYPE
      GO TO 1595
1589  IF (IYPEV(JZ)-NYPE) 1590,1593,1593
1590  IF (IYPEV(JZ)-IYPE) 1591,1595,1594
1591  I=1
      IYPS=IYPEV(JZ)+1
      DO 1592 JY=IYPS,IYPE
      CALL WAKWZ(JZ,JY)
1592  CONTINUE
      IYPEV(JZ)=IYPE
      GO TO 1595
1593  LL=-1
1594  IYPE=IYPEV(JZ)
1595  CONTINUE
      IF (I) 180,180,1585
C
C  ADD POINTS TO THE LEFT SIDE WHERE NEEDED
C
160   IF (IYPSN-2) 170,170,161
161   IYPSN=IYPSN-1
      IF (IYPSN-NYPS) 162,106,106
162   YOLDV(IYPSN-1)=YOLDV(IYPSN)-DFRMX*(YOLDV(IYPSN+1)-YOLDV(IYPSN))
      GO TO 106
C
C  REDUCTION OF NUMBER OF POINTS REQUIRED
C
170   LPKRF=1
180   RETURN
      END
CART ID 0105  DB ADDR 4050   DB CNT 01CC
```

```
WAKSZ.S(0105)
**WAKSZ - STRATIFIED SUBMARINE WAKE, STEP SOLUTION IN Z
      SUBROUTINE WAKSZ
C
C  THIS SUBROUTINE IN THE WAKE PROGRAM MAKES A STEP OF DELTA X/2
C  WITH Y DERIVATIVES EVALUATED AT THE PRESENT X POINT
C
*COPY (CMWAK)
C
C  INITIALIZE COMPUTATION
C
      IOLAY=2
      GO TO (20,10),NSS
10    IROWA=3
      IROWR=2
      MOOD=-1
      LPKRF=0
      GO TO 30
20    IROWA=4
      IROWR=3
      MOOD=0
      IF (LPKRF) 180,30,180
30    NRSTN=NRST
      JENDN=JEND
100   NR=NRSTN-1
      NPOS=NR
      CALL WAKRR(NR,ZV)
      IBOT=IYPS+LYLFF
      ITOP=IYPE
      CALL WAKRR(NR+1,ZPV)
      IBOTP=IYPSP+LYLFF
      ITOPP=IYPEP
      JZ=0
      IF (LZLFF) 101,1015,1015
101   ZM=Z+Z-ZP
      IYPEM=IYPEP
      IYPSM=IYPSP
      GO TO 110
1015  NR=NRSTN
102   JZ=JZ+1
      NPOS=NR
      CALL SFVMV(Z,ZM,NWWZF)
      CALL WAKMR(2,1)
      CALL SFVMV(ZP,Z,NWWZF)
      CALL WAKMR(3,2)
      IBOT=IBOTP
      ITOP=ITOPP
      IF (JZ-JENDN) 104,104,106
104   CALL WAKRR(NR+1,ZPV)
      IBOTP=IYPSP+LYLFF
      ITOPP=IYPEP
      GO TO 110
106   IYPSP=0
      IYPEP=0
      ZP=Z+DFRMX*(Z-ZM)
      ZOLDV(NR+1)=ZP
      IYPSV(NR+1)=0
      IYPEV(NR+1)=0
      IBOTP=0
```

```
      ITOPP=0
C
C  INITIALIZE FOR THE Y DIRECTION AND SWEEP
C
110   Y=YOLDV(IYPS-1)
      YP=YOLDV(IYPS)
      CALL WAKMP(IROWR,NR+1,IYPS-1,XPYV,1)
      CALL WAKMP(IROWT,NR+1,IYPS-1,TPYV,1)
      CALL WAKMP(IROWR,NR,IYPS-1,XV,1)
      CALL WAKMP(IROWT,NR,IYPS-1,TV,1)
      IF (JZ) 111,111,1115
111   CALL WAKRF(XPYV,XMYV,2)
      CALL WAKRF(TPYV,TMYV,4)
      GO TO 112
1115  CALL WAKMP(IROWR,NR-1,IYPS-1,XMYV,1)
      CALL WAKMP(IROWT,NR-1,IYPS-1,TMYV,1)
112   IYPSX=IYPS+LYLFF
      IF (JZ) 1126,1126,1125
1125  CALL WAKMP(IROWA,NR-1,IYPS-1,AV,1)
      CALL WAKMP(IROWG,NR-1,IYPS-1,GM,1)
1126  CALL WAKMP(IROWR,NR+1,IYPS,XPPV,1)
      CALL WAKMP(IROWR,NR,IYPS,XZPV,1)
      CALL WAKMP(IROWT,NR,IYPS,TZPV,1)
      IF (JZ) 113,113,1135
113   CALL WAKRF(XPPV,XMPV,2)
      GO TO 114
1135  CALL WAKMP(IROWR,NR-1,IYPS,XMPV,1)
114   LZERF=0
      IYPSN=MXRY+LYLFF*(MXRY-2)
      IYPEN=1
      DO 145 JY=IYPSX,IYPE
      IF (JY-IYPS) 1141,1142,1142
1141  CALL WAKRF(XPPV,XPMV,1)
      CALL WAKRF(XZPV,XZMV,1)
      CALL WAKRF(XMPV,XMMV,1)
      CALL WAKRF(TZPV,TZMV,3)
      YM=Y+Y-YP
      GO TO 120
1142  IF (JZ) 1145,1145,1144
1144  CALL WAKMP(IROWA,NR-1,JY,AV,1)
      CALL WAKMP(IROWG,NR-1,JY,GM,1)
1145  CALL SFVMV(XPYV,XPMV,NMOVE)
      CALL SFVMV(XV,XZMV,NMOVE)
      CALL SFVMV(XMYV,XMMV,NMOVE)
      CALL SFVMV(TV,TZMV,NMOVE)
      YM=Y
      Y=YP
      YP=YOLDV(JY+1)
      CALL WAKMP(IROWT,NR+1,JY,TPYV,1)
      IF (JZ) 1146,1146,1147
1146  CALL WAKRF(TPYV,TMYV,4)
      GO TO 1148
1147  CALL WAKMP(IROWT,NR-1,JY,TMYV,1)
1148  CALL WAKMP(IROWR,NR+1,JY+1,XPPV,1)
      CALL WAKMP(IROWR,NR,JY+1,XZPV,1)
      CALL WAKMP(IROWT,NR,JY+1,TZPV,1)
      IF (JZ) 1149,1149,1150
1149  CALL WAKRF(XPPV,XMPV,2)
      GO TO 120
```

```
1150  CALL WAKMP(IROWR,NR-1,JY+1,XMPV,1)
C
C  CALCULATE MATRIX COEFFICIENTS AND INVERT FOR GAMMA AND AV
C
120   CALL WAKMZ
      DO 125 I=1,NWVEC
      TEM=1.0/(YMAT(I)-XMAT(I)*GM(I))
      GM(I)=TEM*ZMAT(I)
      AV(I)=TEM*(DVEC(I)-XMAT(I)*AV(I))
125   CONTINUE
      CALL WAKMP(IROWA,NR,JY,AV,2)
      CALL WAKMP(IROWG,NR,JY,GM,2)
      IF (JZ-JENDN) 129,130,130
C
C  UPPER BOUNDARY CONDITION
C
129   IF (JY-IYPSP) 1291,1292,1292
1291  IF (IYPSP-2) 145,145,130
1292  IF (JY-IYPEP) 145,145,130
130   CALL WAKEC(I)
      IF (I) 145,132,145
132   IF (JZ-JENDN) 134,133,133
133   LZERF=1
134   IYPSN=MMIN(JY,IYPSN)
      IYPSN=MMAX(2,IYPSN)
      IYPEN=MMAX(JY,IYPEN)
      IF (JZ+2-MXRZ) 145,170,170
C
C  OUTPUT CURRENT ROW TO DISK
C
145   CONTINUE
      CALL WAKWR(NR,ZV)
      CALL WAKMG(NR,ZV,2)
      IF (JZ-JENDN) 147,146,146
146   IF (LZERF) 1465,150,1465
1465  IYPSP=IYPSN
      GO TO 1485
C
C  ADD POINTS TO THE LEFT SIDE WHERE NEEDED
C
147   IF (IYPSN-IYPSP) 1475,148,148
1475  IYPSP=IYPSN
C
C  ADD POINTS TO THE RIGHT SIDE WHERE NEEDED
C
148   IF (IYPEN-IYPEP) 149,149,1485
1485  IYPEP=IYPEN
149   NR=NR+1
      GO TO 102
C
C  INITIALIZE FOR DOWNWARD PASS
C
150   NR=NR-1
      JENDN=JZ
      IZEND=JENDN-LZLFF-1
      DO 154 IZ=1,IZEND
      NPOS=NR
      CALL SFVMV(Z,ZP,NWWZF)
      CALL WAKMR(2,3)
```

```
      CALL WAKRR(NR,ZV)
      CALL WAKMG(NR,ZV,1)
      IBOT=IYPS+LYLFF
      ITOP=IYPE
      IYPSN=MXRY+LYLFF*(MXRY-2)
      IYPEN=1
      IYEND=IYPEP-IYPSP-LYLFF+1
      DO 1508 IY=1,IYEND
      JY=IYPEP-IY+1
      CALL WAKMP(IROWA,NR+1,JY,AV,1)
      CALL WAKMP(IROWA,NR,JY,XV,1)
      IF (JY-IYPE) 1500,1500,1502
 1500 IF (JY-IYPS) 1501,1506,1506
 1501 IF (IYPS-2) 1506,1506,1502
 1502 DO 1503 I=1,NVAR
      IF (ABS(AV(I)-ZEROV(I))-EPSSV(I)) 1503,1503,1505
 1503 CONTINUE
      GO TO 1508
 1505 IYPSN=MMIN(JY,IYPSN)
      IYPSN=MMAX(2,IYPSN)
      IYPEN=MMAX(JY,IYPEN)
      GO TO 1508
 1506 CALL WAKMP(IROWG,NR,JY,GM,1)
      DO 1507 I=1,NWVEC
      XV(I)=XV(I)-GM(I)*AV(I)
 1507 CONTINUE
      CALL WAKMP(IROWA,NR,JY,XV,2)
 1508 CONTINUE
C
C  ADD POINTS TO THE LEFT SIDE WHERE NEEDED
C
      IF (IYPSN-IYPS) 1515,152,152
 1515 IYPS=IYPSN
C
C  ADD POINTS TO THE RIGHT SIDE WHERE NEEDED
C
 152  IF (IYPEN-IYPE) 153,153,1525
 1525 IYPE=IYPEN
 153  CALL WAKWR(NR,ZV)
      NR=NR-1
 154  CONTINUE
C
C  LOWER BOUNDARY CONDITION
C
      IF (LZLFF) 156,1545,1545
 1545 NR=NR+1
      LZERF=0
      IYPSN=MXRY+LYLFF*(MXRY-2)
      IYPEN=1
      IYEND=IYPE-IYPS-LYLFF+1
      DO 155 IY=1,IYEND
      JY=IYPE-IY+1
      CALL WAKMP(IROWA,NR,JY,AV,1)
      DO 1546 I=1,NVAR
      IF (ABS(AV(I)-ZEROV(I))-EPSSV(I)) 1546,1546,1547
 1546 CONTINUE
      GO TO 155
 1547 IYPSN=MMIN(JY,IYPSN)
      IYPSN=MMAX(2,IYPSN)
```

```
      IYPEN=MMAX(JY,IYPEN)
      LZERF=1
155   CONTINUE
      IF (LZERF) 160,156,160
C
C  RETURN TO MAINLINE
C
156   JEND=JENDN
      NRST=NRSTN
      RETURN
C
C  NEW LOWER ROW REQUIRED
C
160   IF (NRSTN-2) 170,170,161
161   NRSTN=NRSTN-1
      JENDN=JENDN+1
      CALL WAKRR(NR-1,ZMV)
      IBOT=0
      ITOP=0
      IYPSM=IYPSN
      IYPEM=IYPEN
      CALL WAKWR(NR-1,ZMV)
      ZOLDV(NRSTN-1)=ZM-DFRMX*(Z-ZM)
      IYPSV(NRSTN-1)=0
      IYPEV(NRSTN-1)=0
      GO TO 100
C
C  REDUCTION OF NUMBER OF POINTS REQUIRED
C
170   LPKRF=2
180   RETURN
      END
CART ID 0105  DB ADDR 4F20   DB CNT 0216
```

```
WAKTC.S(0105)
**WAKTC - STRATIFIED SUBMARINE WAKE, TURBULENCE CHECK FOR MIN/MAX VALUES
      SUBROUTINE WAKTC(TEMM,TMAX,LFL)
C
C  THIS SUBROUTINE IN THE WAKE PROGRAM CHECKS THAT VV AND WW SATISFY
C  RATIONAL BOUNDS ON THEIR BEHAVIOR IN AN UNSTABLE DENSITY GRADIENT
C
*COPY (CMWAK)
C
      IF (TEMM) 10,100,20
10    TEMM=0.0
      GO TO 40
20    IF (TEMM-TMAX) 100,100,30
30    TEMM=TMAX
40    LFL=LFL+1
100   RETURN
      END
CART ID 0105  DB ADDR 4220   DB CNT 0020
```

```
WAKWR,S(0105)
**WAKWR - STRATIFIED SUBMARINE WAKE, WRITE A Z BUFFER ROW
      SUBROUTINE WAKWR(NRX,ZPOS)
C
C  THIS SUBROUTINE IN THE WAKE PROGRAM WRITES A Z ROW AND
C  FLUSHES THE ROW BUFFER AT THE NRX POSITION FOR ALL Y
C
      DIMENSION ZPOS(2),RECB(24)
*COPY (CMWAK)
      EQUIVALENCE (RECB(1),DVEC(1))
C
      DATA JERRX/2HWR/
C
C  WRITE IYPS AND IYPE VALUES BACK FROM ZPOS
C
      IF (NRX) 300,300,10
10    IF (NRX-MXRZ) 20,20,300
20    CALL SFVMV(ZPOS,ZA,NWWZF)
      ZOLDV(NRX)=ZA
      IYPSV(NRX)=IVECA
      IYPEV(NRX)=IVECB
C
C  LOCATE NRX POSITION IN ROW BUFFER ARRAY
C
      J=NRX-NPOS+2
      IF (J) 300,300,30
30    IF (J-3) 40,40,300
40    IF (IVECA) 150,150,50
50    IYPSX=IVECA+LYLFF
      IYPEX=IVECB
C
C  WRITE NRX INFORMATION FROM BUFFER ROW
C
      NRXX=(NRX-1)*MXRY+IYPSX
      DO 130 I=IYPSX,IYPEX
      CALL PBFDR(SLNID,NRXX,NWR,RECB)
      CALL SFVMV(ROWB(1,I,J),RECB(1),NWVEC)
      IF (MOOD) 105,110,115
105   CALL SFVMV(ROWB(7,I,J),RECB(7),NMOVE)
      GO TO 120
110   CALL SFVMV(ROWB(7,I,J),RECB(13),NMOVE)
      IF (I-IBOT) 112,111,111
111   IF (I-ITOP) 120,120,112
112   CALL SFVMV(ZEROV,RECB(7),NWVEC)
      GO TO 120
115   CALL SFVMV(ROWB(7,I,J),RECB(7),NWVEC)
      CALL SFVMV(ROWB(13,I,J),RECB(19),NWVEC)
120   NRXX=NRXX-1
      CALL PBFDW(SLNID,NRXX,NWR,RECB)
130   CONTINUE
150   RETURN
300   JERR=JERRX
      RETURN
      END
CART ID 0105  DB ADDR 3040   DB CNT 006A
```

```
WAKWZ.S(0105)
**WAKWZ - STRATIFIED SUBMARINE WAKE, WRITE A ZERO POINT
      SUBROUTINE WAKWZ(NRZ,NRY)
C
C  THIS SUBROUTINE IN THE WAKE PROGRAM WRITES
C  A ZERO POINT TO THE WORKING FILE
C
      DIMENSION RECB(24)
*COPY (CMWAK)
      EQUIVALENCE (RECB(1),DVEC(1))
C
      DATA JERRX/2HWZ/
C
      IF (NRZ) 100,100,10
10    IF (NRZ-MXRZ) 20,20,100
20    NRXX=(NRZ-1)*MXRY+NRY
      CALL SFVFL(0.0,RECB,NWVEC)
      DO 30 I=7,NWR,NWVEC
      CALL SFVMV(ZEROV,RECB(I),NWVEC)
30    CONTINUE
      CALL PBFDW(SLNID,NRXX,NWR,RECB)
      RETURN
100   JERR=JERRX
      RETURN
      END
CART ID 0105  DB ADDR 5140   DB CNT 0030
```

```
WAKTM.S(0101)
*IOCS(2501 READER,DISK)
**INITIALIZATION PROGRAM FOR FULL PLANE SWIRL IN WAKE PROGRAM
      DEFINE FILE 1(32000,2,U,NR)
      DIMENSION YOLDV(40),ZOLDV(40),VEC(40)
      NINU=8
      READ (NINU,1000) NY
1000  FORMAT(I4)
      READ (NINU,1001) (YOLDV(JY),JY=1,NY)
1001  FORMAT(E12.4)
      RS=.4
      RV=1.5
      NZ=NY+NY-1
      DO 10 JY=1,NY
      IY=NY-JY+1
      IZ=NZ-JY+1
      YOLDV(IY)=RS*YOLDV(IY)
      ZOLDV(JY)=-YOLDV(IY)
      ZOLDV(IZ)=-ZOLDV(JY)
10    CONTINUE
      NY=NZ
      CALL SFVMV(ZOLDV,YOLDV,NY)
      RMAX=ZOLDV(NZ)
      NR=1
      RR=0.0
      WRITE (1'NR) RR
      WRITE (1'NR) NY,NZ,(YOLDV(JY),JY=1,NY),(ZOLDV(JZ),JZ=1,NZ)
      DO 100 I=1,6
      DO 90 JY=1,NY,10
      IY=MMIN(JY+9,NY)
      DO 80 JZ=1,NZ
      CALL SFVFL(0.0,VEC,40)
      Z=ZOLDV(JZ)
      IF (JZ-1) 70,70,30
30    IF (JZ-NZ) 40,70,70
40    DO 60 J=JY,IY
      Y=YOLDV(J)
      IF (J-1) 60,60,42
42    IF (J-NY) 45,60,60
45    R=SQRT(Y*Y+Z*Z)
      IF (R-RMAX) 46,60,60
46    GO TO (51,60,52,54,54,60),I
51    RR=(Z*Z+Y*Y)/RS/RS
      VEC(J)=.0108/(1.0+RR/6.25)**2
      GO TO 60
52    RR=0.5*(Z*Z+Y*Y)/RS/RS
      VEC(J)=0.080*(1.0-RR)*EXP(-RR)
      GO TO 60
54    RR=3.0*(Y*Y+Z*Z)
      IF (R) 60,60,55
55    IF (R-RV) 555,60,60
555   VEL=(1.0-EXP(-RR))*(RV-R)**2/R/15.0
      IF (I-4) 56,56,57
56    VEC(J)=-VEL*Z/R
      GO TO 60
57    VEC(J)=VEL*Y/R
60    CONTINUE
70    WRITE (1'NR) (VEC(J),J=JY,IY)
80    CONTINUE
```

```
90    CONTINUE
100   CONTINUE
      RR=-1.0
      WRITE (1'NR) RR
      CALL EXIT
      END
CART ID 0101  DB ADDR 5410   DB CNT 0080
```

IBM

FORTRAN Coding Form

GX28-7327-6 U/M 050**
Printed in U.S.A.

PROGRAM		PUNCHING INSTRUCTIONS	GRAPHIC		PAGE OF
PROGRAMMER	DATE		PUNCH		CARD ELECTRO NUMBER*

STATEMENT NUMBER — CONT — FORTRAN STATEMENT — IDENTIFICATION SEQUENCE

```
   1   1  10
      FULL PLANE SWIRL WITH AXIAL VELOCITY AT FR = 10
 300   0.0      50.8    .01       24   0
  40   0  40   0
   1   1   1   0   1   1   1   1   0   1   1   0   1   1
.0001   1.      .1      .01     .01
.5      1.      .5      .0      .0
.0      .0      .0      .0      .0      .0      .0
.001    .05     .001    .02     .05
.63     .00001  1.5     2.      1.1     .05     .05     2.02
   1   0  10 500  20   0   2   0
19.4    50.8
   3   5   4   1   0   0   5   1   1   1   1   1   0   1
3.68E+07.01     1.      2.5     .75     .125    .0      .1      1.8
.3      1.      .25     -1.     .2      .005    .01     100.    .005
-.35    -.6     .8      .375    .375    .75         0
.001    .1      2.      1.      40.     .0001   .5      6.      .0      .0
WAKWF   WAKGM   WAKGL   WAKPL
         3    -1   1    0    1
```

*A standard card form IBM electro 888157 is available for punching statements from this form

**Number of forms per pad may vary slightly

A-94